DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Oh

19950206 093

Selecting Optimal Experiments
for
Feedforward Multilayer Perceptrons

DISSERTATION

Lisa M. Belue
Captain, USAF

AFIT/DS/ENS/95-01

DTIC QUALITY INSPECTED 4

Approved for public release; distribution unlimited

AFIT/DS/ENS/95-01

Selecting Optimal Experiments

for

Feedforward Multilayer Perceptrons

DISSERTATION

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy
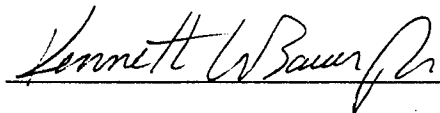
Lisa M. Belue, B.S., M.S.

Captain, USAF

March, 1995

Approved for public release; distribution unlimited

Selecting Optimal Experiments
for
Feedforward Multilayer Perceptrons
Lisa M. Belue, B.S., M.S.
Captain, USAF

Approved:

_____   11 JAN 95
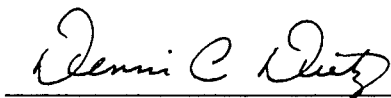Lt Col Kenneth W. Bauer, Jr., Chairman   Date

_____   11 Jan 95
Dr Steven K. Rogers, Committee Member   Date

_____   11 Jan 95
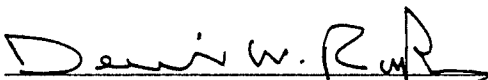Lt Col Dennis C. Dietz, Committee Member   Date

_____   11 JAN 95
Capt Dennis W. Ruck, Committee Member   Date

_____   11 Jan 95
Prof Daniel E. Reynolds, Dean's Representative   Date

Accepted:

_____   25 Jan. 1995
Dr. J. S. Przemieniecki, Senior Dean   Date

ii

Preface

The goal of this research was to develop a method for selecting optimal training vectors for multilayer perceptrons in an experimental setting. This has been accomplished. The resulting strategy can be implemented whenever experiments are to be performed and a multilayer perceptron will be used to model the data.

My success in this goal is the result of the assistance, guidance and support of many people. I wish to acknowledge my indebtedness to my committee chairman Lt Col Kenneth W. Bauer, Jr. for his support, enthusiasm and optimism. I also want to thank my other committee members, Dr. Steve Rogers, Lt Col Dennis Dietz, Capt Dennis Ruck, and the Dean's representative Professor Daniel Reynolds for their comments and suggestions. Wright Laboratory's Survivability Enhancement Branch and specifically Ms Pat Pettit, sponsored this research. They provided me with the computer, the data, and a constant reminder that the work I was doing might really matter.

I don't think its possible to complete an endeavor of this magnitude without the comraderie of one's peers. A special thanks goes to Jean Steppe. The credit for numerous ideas in this research should be shared with her. Also, thanks goes to Dan Zalewski. His computer expertise is unequaled and the times he let me ramble on about my research was a huge help. I would also like to acknowledge the good work done by the people who keep things running in ENS—Jackie Logan and Nancy Freese.

I am grateful for my all-weather running partners, Col Parnell and Major Lehmkuhl for helping me keep the big-picture perspective. May your shoes never be wet.

Most important of all is the support of my family. I would like to thank my husband Ken and daughter Caitlin for their encouragement—they cheered me up when I needed it and let me work when I need to. I never doubted their support.

<div align="right">Lisa M. Belue</div>

Table of Contents

List of Figures

List of Tables

Abstract

Where should a researcher conduct experiments to provide training data for a multilayer perceptron? This question is investigated and a statistically-based method for optimally selecting experimental design points for multilayer perceptrons is introduced. Specifically, a criterion is developed based on the size of an estimated confidence ellipsoid for the weights in the multilayer perceptron. This criterion is minimized over a set of exemplars to find optimal design points. Until now, only graphical and heuristic algorithms were available.

Initially, single output networks are examined in which the multilayer perceptron is viewed as a univariate nonlinear model. An example is used to demonstrate the superiority of optimally selected design points over randomly chosen points and points chosen in a grid pattern. Also, two measures are successfully used to rank the design points in terms of their importance. Due to the dense interconnectivity of multilayer perceptrons, locating design points can be computationally complex. Therefore, two methods are presented as avenues to significantly reduce complexity—a *distributed linear feedthrough network structure* and a *weight subset method.*

Next, multiple output networks are examined with the multilayer perceptron viewed as a multivariate nonlinear model. The criterion for selecting design points in this framework becomes more complex and a simplifying technique is employed to judiciously choose desired outputs of the network to produce uncorrelated actual outputs.

Finally, the methods described above are integrated into a comprehensive procedure and are tested on two applications dealing with aircraft survivability. The single output methodology is demonstrated on the classification of the performance of armor piercing incendiary projectiles striking composite materials and the multiple output methodology is applied to a seven-class problem relating time and stress to the performance of the projectiles. In both cases, simulating the indicated experiments produced a superior multilayer perceptron.

Selecting Optimal Experiments

for

Feedforward Multilayer Perceptrons

I. Introduction

The objective of this research is to develop a cohesive strategy for selecting design points for experimentation so as to develop accurate multilayer perceptron classifiers. Only recently have neural networks come to the forefront of discriminant analysis methods. Neural networks, and more specifically multilayer perceptrons, have the advantage of learning their optimal parameters and are simple to apply once these parameters are found. In addition, multilayer perceptrons have very general functional forms that can be expanded or contracted to suit the current application. A further advantage in a discrimination setting is that multilayer perceptrons allow for the formation of nonlinear decision regions, including disjoint regions. Finally, there is the appealing quality of the "brain-like" structure of neural networks that has caused attention to be turned to these relatively new classifiers. Although Rosenblatt developed the perceptron as early as 1957, a paper published in 1969 by Minsky and Pappert highlighting the inadequacies of the perceptron stifled neural network research for several years. It wasn't until 1982 that, to a large degree, Hopfield sparked a resurgence in this area [32, 55].

Optimal experimental design seeks to select vectors from some region of operability such that the design defined by these vectors is in some sense optimal. Most often this problem consists of developing some sensible criterion based on an assumed model and using this criterion to obtain a design. In contrast to the neural network arena, primary research on optimal experimental design was performed over thirty years ago. In 1943 Wald proposed the

1

determinant criterion and in 1959, Box and Lucas demonstrated how this criterion could be applied to nonlinear models [58].

This research explores the synthesis of these two topic areas and, where possible exploits the flexibility and dense interconnectivity of multilayer perceptrons. The following example sets the stage for the research covered in succeeding chapters:

> A researcher (chemist, engineer, bio-technician, etc.—whichever is most familiar) believes that a system he is investigating would be best modeled by a multilayer perceptron with some set of physical characteristics of the system as inputs. A small amount of screening data is available. The researcher wishes to perform additional experiments to develop the most accurate multilayer perceptron classifier possible. Which experiments should he perform?

This chapter briefly outlines the importance of experimental design and introduces multilayer perceptrons including their structure, training methods and other practical considerations. The chapter concludes with a statement of dissertation research objectives and an overview of remaining chapters.

1.1 Importance of Experimental Design

The reason that experimental design is important is that "the information content of the data is established when the experiment is performed" and no amount of innovative data analysis can recover information which is not present in the data [66]. Figure 1 shows how experimental design fits into the overall process of developing multilayer perceptron classifiers. First, an initial model is proposed and experiments performed with data collected. Using this initial model, the multilayer perceptron is trained using the collected data. Next, the multilayer perceptron is evaluated in terms of existing data to determine first whether the model is appropriate and second, whether further data is required. If the model continues to be appropriate, an experimental design scheme is employed and used to collect further data. Notice that the form of the multilayer perceptron influences the experimental design.
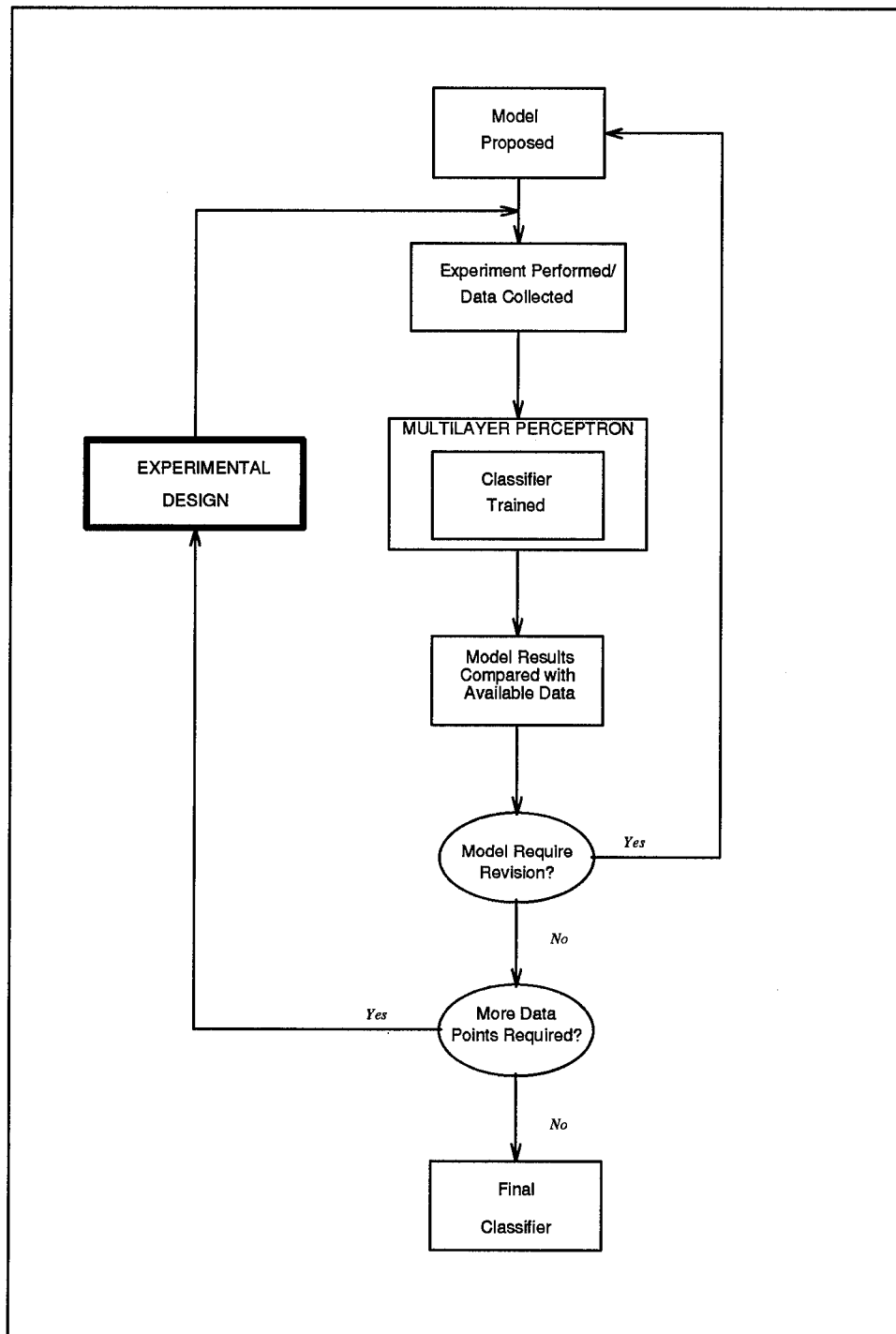
Figure 1.   Experimental Design in the Overall Development of Multilayer Perceptron Models
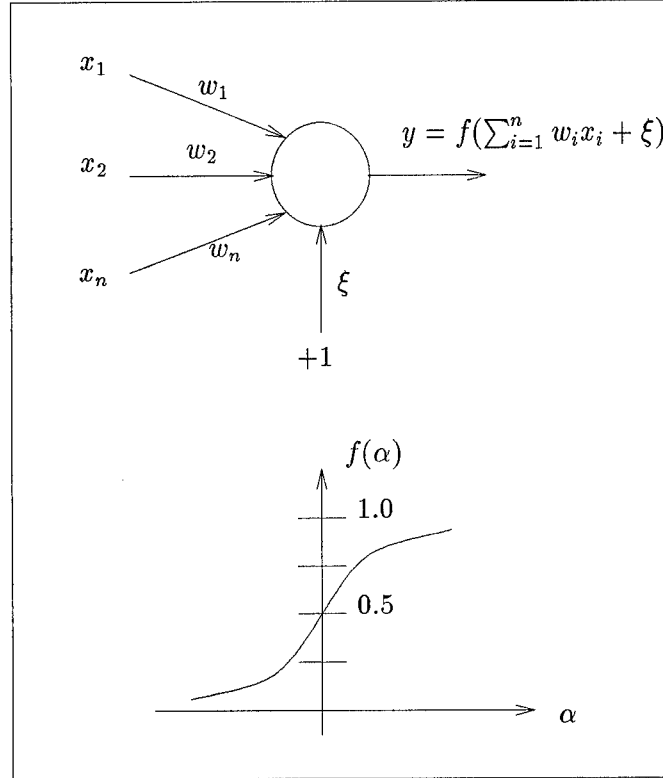
Figure 2. Single Perceptron

Reprinted from [55]

1.2 Feedforward Multilayer Perceptrons

1.2.1 Structure. Figure 2 shows a single perceptron. (Appendix D lists the symbols used.) Data feeds into the perceptron's input nodes numbered 1 to n with input values x_1 to x_n and the w_i on each branch of the perceptron weight the inputs. The procedure sums across the weighted inputs, adds a bias term, and transforms the sum so that the activation y of the perceptron is:

$$y = f[(\sum_{i=1}^{n} w_i x_i) + \xi] \tag{1}$$

The bias is an additional node whose input is 1. Therefore, the bias times the weight connecting the bias (ξ) is a constant. The nonlinear transformation $f[\cdot]$ most often takes the

4

form of a sigmoid:

$$f(\alpha) = \frac{1}{1 + e^{-\alpha}} \tag{2}$$

For each input, the perceptron outputs a single value that signifies the classification of the input [55]. Training the perceptron to classify inputs consists of finding the weights that produce outputs near certain desired values. Most often, desired outputs are set to 1 or 0 to denote class membership.

The single layer perceptron does not allow for discrimination between classes that are not hyperplane separable [43]. Beginning in the 1980's, researchers developed methods for layering the single perceptron to allow for complex, nonlinear boundaries between classes [57]. Figure 3 shows a two layer perceptron. Cybenko has shown "only one hidden layer is sufficient for any arbitrary transformation, given enough nodes" [55].

The input layer of a multilayer perceptron will have as many nodes as there are features plus an additional node for the bias term. The output layer will normally have one node for every class of outputs. Consequently, the structure of multilayer perceptrons varies only in the number of hidden layers and the number of nodes within each of these hidden layers. Although methods have been suggested to determine the best structure [17, 31, 33, 38], as Ruck states: "Rigorous mathematical techniques have not been developed to determine the appropriate number of hidden layers or the number of nodes in those layers for a given problem" [57]. For the most part, a trial and error approach is taken.

1.2.2 Training. Training algorithms are rules by which the perceptron will update weights (learn) as the user presents data. Backpropagation is the most prevalent method for updating the weights in a multilayer perceptron. This algorithm is a gradient descent method for training the weights in a multilayer perceptron while minimizing the mean squared error between the outputs of the network and the desired outputs [40].

In a multilayer perceptron, the data is introduced to the input layer and propagated through the network in a feedforward manner. Comparing the output of the perceptron with

5

NETWORK

z_1 z_2 z_r Outputs

...

w_{ij}^2 ξ_j^2

Hidden
Layer

+1

w_{ki}^1 ξ_i^1

...

x_1 x_2 x_n +1 Inputs

DETAIL OF MIDDLE NODE

$$x_2^1 = f(\sum_{k=1}^n x_k w_{k2}^1 + \xi_2^1)$$

ξ_2^1 +1

w_{12}^1 w_{22}^1 w_{n2}^1

...

x_1 x_2 x_n
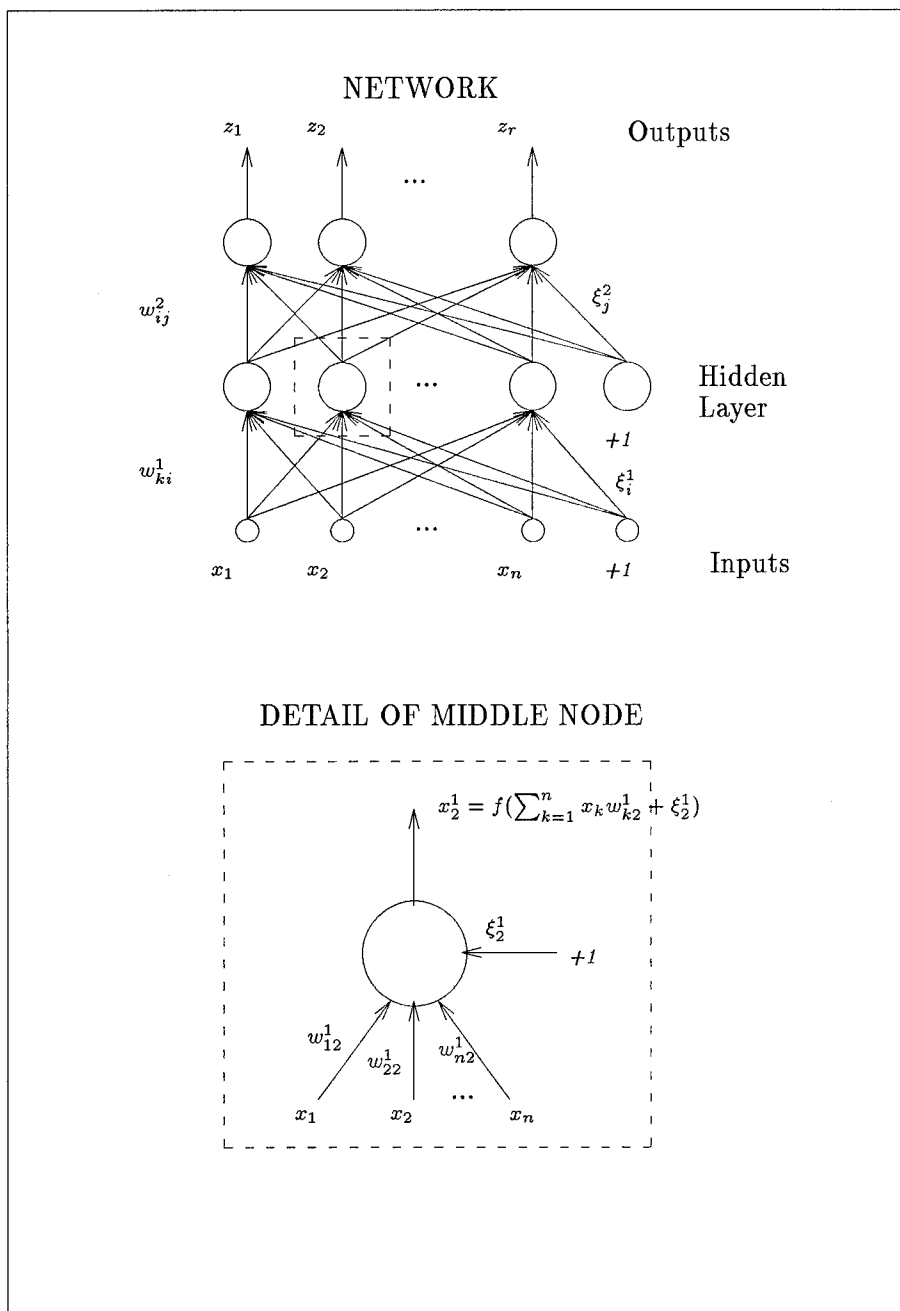
Figure 3. Multilayer Perceptron

6

the desired classification yields an error term used to compute a correction for the weights [55]. Listed below is the backpropagation training method.

Backpropagation Training

1. Initialize weights and biases to small random values.

2. Present training input and desired outputs.

3. Calculate outputs.

4. Adapt weights and biases according to

$$w_{ij}^+ = w_{ij}^- + \eta \delta_j x_i + \alpha(w_{ij}^- - w_{ij}^{--}) \qquad (3)$$

where w_{ij} is the weight from node i to node j in the next layer, x_i is the output of node i, and δ_j is the *error* associated with node j. η is the learning rate and α is the momentum rate (for example constants of .35 and .7 respectively). w_{ij}^+ is the new weight value and w_{ij}^- is the old weight value. w_{ij}^{--} is the value of the weight before the last update. Thresholds are adapted similarly where x_i is replaced by +1 if the bias is *added* to the weighted sum and -1 if it is *subtracted*. The δ_j are defined as follows:

$$\delta_j = \begin{array}{ll} z_j(1-z_j)(d_j - z_j) & \text{for output node } j \\ x_j(1-x_j)\sum_k \delta_k w_{jk} & \text{for hidden node } j \end{array} \qquad (4)$$

where d_j is the desired output for output node j and z_j is the actual output. For hidden nodes the δ_k are the errors for the layers above.

Often the input vectors must be normalized in some fashion so that no one feature dominates the classification process. Normalization can be accomplished by scaling the features in each vector to values between 0 and 1 based on the range of the values of the features in the training set.

Available data is randomly assigned to one of three sets [28]:

- The Training Set: This set of feature vectors is presented to the multilayer perceptron for training. These vectors contain the desired classification of the feature vector.

7

- The Test Set: The test set is used to test the accuracy of training while training is ongoing. After each epoch (i.e., each complete presentation of the training set), each test set vector is presented to the network and classified. This classification is then compared to the desired classification, and an error computed. These test vectors act as controls for determining when the accuracy of the perceptron is at an acceptable level.

- The Validation Set: After the multilayer perceptron is considered optimally trained, the validation set vectors are presented, classified and that classification compared to the true classification of each vector. This set acts as a verification of the performance of the classifier since its vectors are not directly used during the classifier's development.

The output error is observed during the training of the multilayer perceptron to judge the accuracy of the discriminator at any epoch. The output error ($\mathcal{E}_O$) is defined as:

$$\mathcal{E}_O = \sum_{i=1}^{r} \sum_{s=1}^{N} (rN)^{-1} \mid (d_i^s - z_i^s) \mid \tag{5}$$

where r is the number of output nodes (also the number of classes), N is the number of exemplars (observations) in the set of interest, d_i^s is the desired output for the ith node when the sth exemplar is presented, and z_i^s is the actual output of node i for the sth exemplar. In other words, the output error is the average absolute amount that the output of the network differed from the desired output. Classification error ($\mathcal{E}_C$) is also used to judge the accuracy of the multilayer perceptron during training. Classification error is defined as the percentage of the vectors in the data set of interest that are classified incorrectly. If ω^s is defined as the true classification of feature vector s and $\hat{\omega}^s$ as the actual classification of feature vector s and let

$$I_i(\omega) = \begin{cases} 1 & \text{if } \omega \in \text{Class } i \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

then

$$\mathcal{E}_C = 100 \left[1 - N^{-1} \sum_{i=1}^{r} \sum_{s=1}^{N} (I_i(\omega^s) \cdot I_i(\hat{\omega}^s)) \right] \tag{7}$$

The method for training a multilayer perceptron is illustrated in Figure 3. First, the initial weights in the perceptron are randomized within a range typically between -0.5 and 0.5. This is done because even under identical learning conditions, random initial weights can lead

8

to results that differ from one training session to another. Next, an epoch begins by presenting a randomly selected training exemplar. The order that training vectors are presented is also randomized. Using the current values of the weights, the output of the network is calculated. The next step is to compare the actual output to the desired output.

Based on this comparison, all the weights in the network are updated. Feature vectors from the training set are presented to the network until all vectors have been presented once. The weights are adjusted after each training vector presentation. After all the training vectors have been used, the weights are fixed and the vectors in the test set are presented. $\mathcal{E}_O$ and $\mathcal{E}_C$ are calculated. If the rates are acceptable, the multilayer perceptron is considered trained and the validation set is presented for a final analysis of the error rate. If the error rates are unacceptable, then another epoch begins and all the training data is presented again with weights being updated after the introduction of every training vector. It is not unusual for thousands of epochs, i.e., presentations of the training data, to be necessary to achieve an acceptable error rate.

When several runs of the same network architecture require comparison, typically an average error rate $\bar{\mathcal{E}}$ is calculated. A confidence interval around the true mean error provides information as to the variability of the result. For a reasonably large number of runs, M, the t-distribution can be used for confidence interval estimation. In this case, the confidence interval for the expected error rate takes the form

$$\bar{\mathcal{E}} - t_{(1-\frac{\alpha}{2};M-1)}\frac{s}{\sqrt{M}} < \mu < \bar{\mathcal{E}} + t_{(1-\frac{\alpha}{2};M-1)}\frac{s}{\sqrt{M}} \tag{8}$$

where $t_{(1-\frac{\alpha}{2};M-1)}$ is determined by the t-distribution for confidence coefficient $1 - \alpha$ and $M - 1$ degrees of freedom and [42:343,374]

$$\bar{\mathcal{E}} = M^{-1}\sum_{i=1}^{M}\mathcal{E}_i \tag{9}$$

9
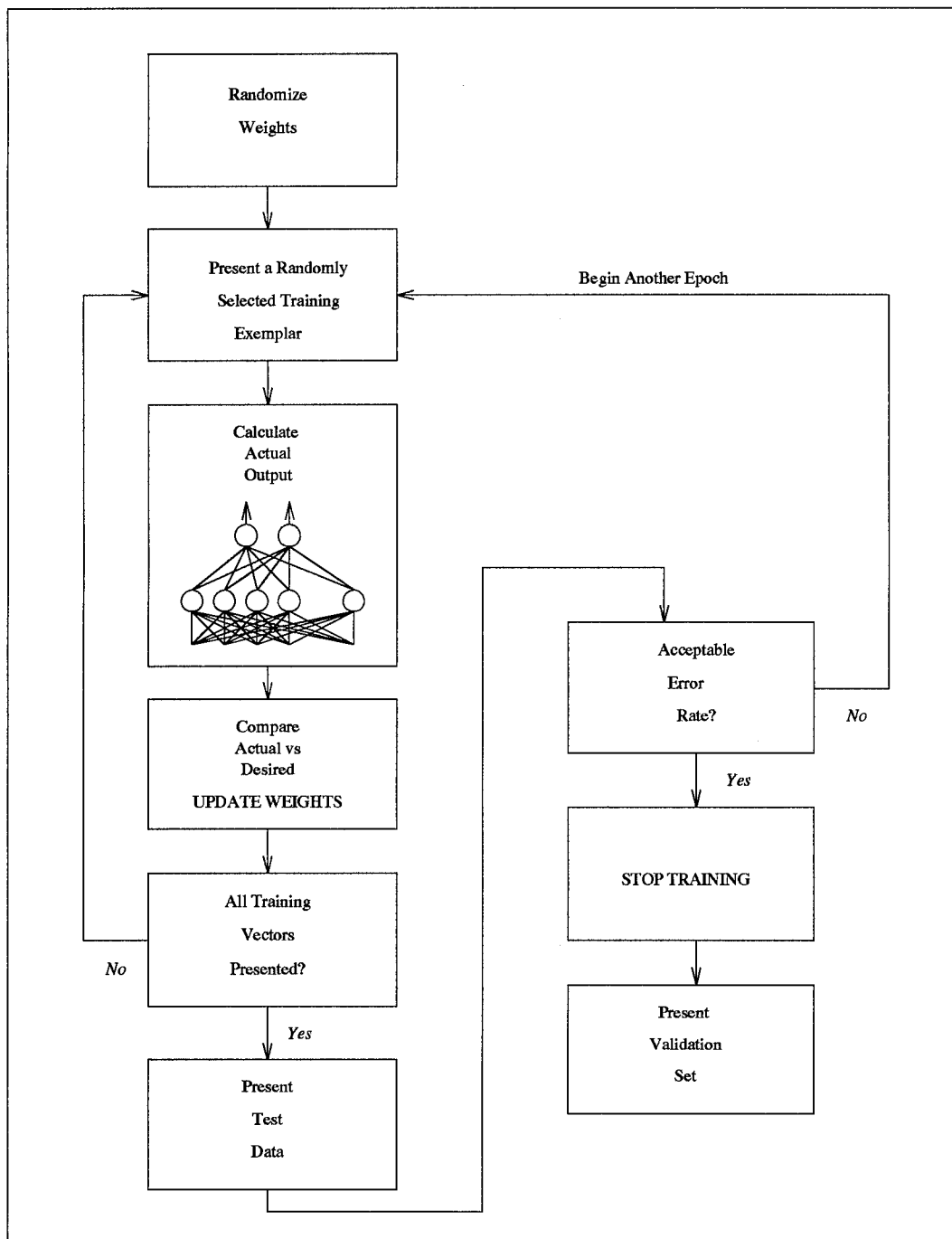
Figure 4. Training Procedure for Multilayer Perceptron

10

Table 1. Hypothesis Test for Equality of Means

H_o	$\mu_1 = \mu_2$
H_a	$\mu_1 \neq \mu_2$
Test Statistic	$T = \dfrac{\bar{\mathcal{E}}_1 - \bar{\mathcal{E}}_2}{S\sqrt{\frac{1}{M_1} \frac{1}{M_2}}}$ $S = \sqrt{\dfrac{(M_1-1)s_1^2 + (M_2-1)s_2^2}{M_1+M_2-2}}$
Rejection Region	$\|t\| > t_{(1-\frac{\alpha}{2};M_1+M_2-2)}$
Assumptions	$\mathcal{E}_{ij}$ are independent samples from normal distributions with $\sigma_1^2 = \sigma_2^2$

$$s^2 = (M-1)^{-1} \sum_{i=1}^{M} (\mathcal{E}_i - \bar{\mathcal{E}})^2 \qquad (10)$$

This confidence interval is based on the assumption that the sample has been randomly selected from a normal population. It is appropriate for samples of any size and works well as long as departures from normality are not excessive [42:373].

When methodologies require comparison, one typically makes several runs of the multilayer perceptron for each of the methods. A hypothesis test can be conducted to determine whether the average error rates for the methods are significantly different. Let M_i be the number of runs for method i ($i = 1, 2$), $\bar{\mathcal{E}}_i$ be the average error rate for method i, μ_i be the true error rate and define

$$\bar{\mathcal{E}}_i = M_i^{-1} \sum_{j=1}^{M_i} \mathcal{E}_{ij} \qquad (11)$$

$$s_i^2 = (M_i - 1)^{-1} \sum_{j=1}^{M_i} (\mathcal{E}_{ij} - \bar{\mathcal{E}}_i)^2 \qquad (12)$$

Then, the test for comparing two means is listed in Table 1 [42:457].

Figure 5 summarizes a procedure which can be used to iterate through the possible multilayer perceptron architectures and settings to arrive at an optimal network. Initially, the number of middle nodes, the learning rate and the momentum rate are set to values that are suggested by the discrimination problem and by the user's experience. The network is trained and the number of epochs increased until the minimum test set error is observed. The number of middle nodes is increased as long as the minimum test set error continues to decrease, or fewer number of epochs are required to achieve the minimum. Next, the number of middle nodes is fixed and the learning and momentum rates are tested over an applicable range. This range of learning and momentum rates may depend on the order of the discrimination problem (i.e., how many feature inputs/exemplars are involved) and the observed behavior of the error as these rates are changed. After the multilayer perceptron is tested over the range of learning and momentum rates, the rates yielding the lowest test set error rate are chosen.

1.3   Feature Extraction—Saliency Metrics

1.3.1   Background. When employing neural networks of any type, one objective is to limit the number of input features. Devijver and Kittler cite "the curse of dimensionality" as the primary reason for limiting these features [18]. The dilemma is that as the number of features increases, the number of training vectors required in the training set also increases. Foley states that the ratio of training vectors per class to the feature size should be greater than three. He claims that satisfying this condition ensures that the test set error rate is close to the true error rate [24]. Work by Cover reinforces the result that more features require more training data [16].

Intuitively an analyst would like to include only those features that make a significant contribution to the network. When designing a classifier, the features that provide information should be included and those that provide little information should not be included as inputs to the network. To insure optimal feature extraction, exhaustive enumeration of the feature sub-
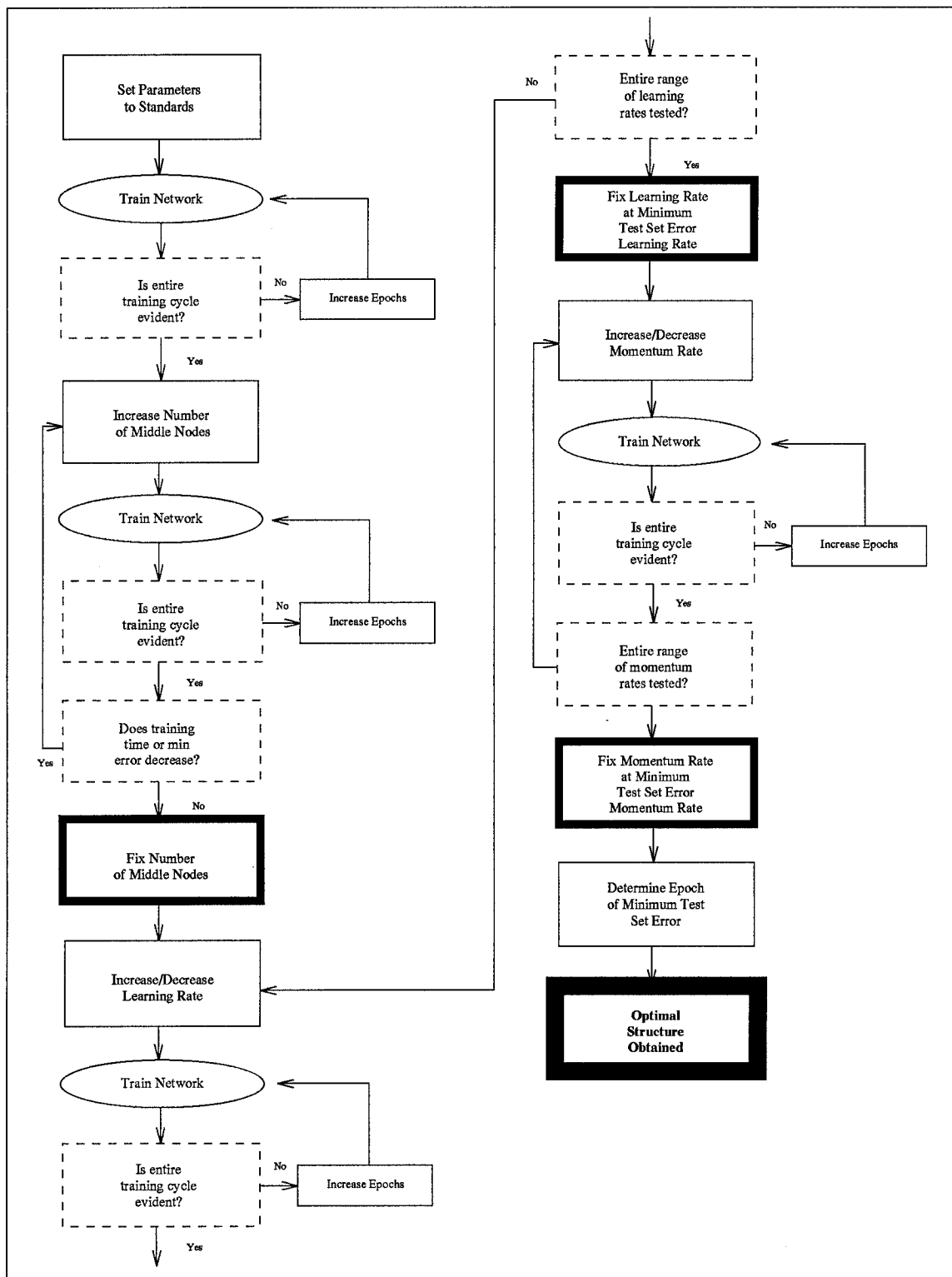
Figure 5. Procedure for Determining Multilayer Perceptron Structure

13

sets could be applied [18]. In a situation with very many variables, this would be impractical. Ruck develops a measure for ranking features called a "saliency metric" [56]. To produce the saliency measure, "the sensitivity of the network's output to its input is used to rank the input feature's usefulness" [56]. The saliency measure allows for the examination of the inputs as they relate to the output values of the multilayer perceptron without examining every subset.

The application of the saliency measure begins with the calculation of the derivative of the output with respect to a specific input. When the sigmoid nonlinearity is used for a network with a single hidden layer, this derivative is

$$\frac{\partial z_j}{\partial x_k} = z_j(1 - z_j) \sum_i w_{ij}^2 \delta_i^1 w_{ki}^1 \tag{13}$$

where z_j is the output of node j in the output layer, w_{ij}^2 is the weight connecting the hidden layer with the output layer, w_{ki}^1 is the weight connecting the input layer with the hidden layer and $\delta_i^1 = x_i^1(1 - x_i^1)$ where x_i^1 is the output of node i in layer 1. (Reference [57] for a detailed derivation.) Each of the pieces that make up the saliency measure are illustrated in Figure 3. From the equation above, it is apparent that the derivative depends on the inputs to the network as well as the weights within the network.

Ideally, the feature space would be sampled over the entire range of possible values. Letting R be the number of sampling points for each input and letting n be the number of features, then the total number of derivative calculations is R^n which may be excessively large. Ruck suggests that instead of sampling all points, each feature should be sampled over its range *while the other feature inputs are held constant at their actual values* ignoring interactions among the inputs. If there are N training vectors, this method results in RNn derivative calculations [57].

14

Let ψ^m be the vector of R uniformly spaced points covering the range of the mth input feature, then the ith component, ψ_i^m can be defined as

$$\psi_i^m = \min x_m + (i-1)\frac{\max x_m - \min x_m}{R-1} \quad i = 1, \ldots, R \qquad (14)$$

where $\min x_m$ ($\max x_m$) is the minimum (maximum) value of x_m taken over all N training vectors. Finally, Ruck's saliency measure for feature input k is defined as

$$\Lambda_k = \sum_{s=1}^{N} \sum_{k=1}^{n} \sum_{i=1}^{R} \sum_{j=1}^{r} \left| \frac{\partial z_j}{\partial x_k}(\mathbf{x}_s^{k(i)}, \mathbf{w}) \right| \qquad (15)$$

where N is the number of training vectors; n is the number of features; R is the number of uniformly spaced points covering the range of each input feature; r is the number of output classes; the vector $\mathbf{x}_s^{k(i)}$ is the vector $\mathbf{x}_s$ with its kth component replaced by ψ_i^k and $(\mathbf{x}_s^{k(i)}, \mathbf{w})$ indicates that the derivative is evaluated with the feature vector $\mathbf{x}_s^{k(i)}$ and the final estimate of the trained network weight parameter $\mathbf{w}$ [53].

A simpler method of determining the relative significance of the input features once the network has been trained has been suggested by Tarr [63]. He proposes the following alternate saliency measure for a feature input k:

$$\tilde{\Lambda}_k = \sum_i (w_{ki}^1)^2 \qquad (16)$$

Which is simply the sum of the squared weights between the input layer and the first hidden layer.

To understand the relationship between these two saliency measures, it is necessary to examine a simplified form of Ruck's saliency measure. The simplified saliency measure, $\hat{\Lambda}_k$, does *not* examine points in the feature space other than points given in the training and test

data.

$$\hat{\Lambda}_k = \sum_{x \in X} \sum_j \mid \frac{\partial z_j}{\partial x_k}(\mathbf{x}, \mathbf{w}) \mid \qquad (17)$$

where X is the set of all exemplars in the training set and j as before is the index for the output nodes.

Steppe has shown that this simplified measure can be bounded above by an expression containing a constant vector, independent of the feature under study and a term dependent on the weights connected to the feature under study.

$$\hat{\Lambda}_k \leq |\mathbf{w}_k^1| \Pi \qquad (18)$$

where $|\mathbf{w}_k^1|$ is an $1 \times m$ vector of the absolute values of the weights connecting input feature k to middle nodes 1 through m and Π is a $m \times 1$ vector of constants for middle nodes 1 to m. The vector of constants, Π, is identical for each feature being examined. Therefore, an examination of some norm of $|\mathbf{w}_k^1|$ provides a measure of saliency for feature k [61].

1.3.2 Determining Salient Features. Currently, the saliency measures discussed are calculated and features are included subjectively, based on rank order according to the average saliency measures over several training cycles. A method is needed which takes into consideration the saliency of a feature relative to the saliency of a known irrelevant feature. To establish a working procedure for determining which features are significant, a noise variable is included as a feature input along with the original inputs to represent an absolutely insignificant piece of information. Because all continuous features were normalized between zero and one, the added noise was taken as random samples from a uniform (0,1) distribution. The procedure for determining significant feature inputs when a noise feature is present is developed by Belue and Bauer and outlined below [7].

16

Determining Significant Features with Injected Noise

1. Introduce a noise feature to the original set of feature vectors.

2. Train the network.

3. Compute the saliency of all features (using either Λ or $\tilde{\Lambda}$).

4. Repeat steps 2 and 3 at least 30 times (with weights being randomly initialized and training and test sets being randomly selected at the beginning of each training cycle).

5. Assume the average saliency of noise is normally distributed and find the upper one-sided $(\alpha \times 100)$ percent confidence interval for the mean value of the saliency of noise.

6. Choose only those features whose average saliency value falls outside this confidence interval.

7. Retrain the network with the salient features.

Steppe observed that Bonferroni hypothesis testing is appropriate since a "family" of tests are being performed, rather than an individual hypothesis test. In addition, she noted that since the saliency observations are paired and dependent, a paired t-test should be used [61].

1.4 Research Objectives

The objective of this research is to

1. Develop sampling methods for *single output* multilayer perceptrons to select design points for experimentation so as to best estimate the multilayer perceptron parameters.

2. Develop sampling methods for *multiple output* multilayer perceptrons to select design points for experimentation so as to best estimate the multilayer perceptron parameters.

3. Integrate the methods developed into a cohesive strategy for selecting design points.

4. Test this strategy on practical, realistic discrimination problems.

1.5 Research Overview

This remainder of this dissertation is organized into four chapters. Chapter II introduces nonlinear regression and outlines methods for determining experimental designs for input

exemplars. In Chapter III, methods for developing designs for single output multilayer perceptrons are presented. In Chapter IV, results are expanded to include the multiple output case. Finally, Chapter V presents the overall methodology and applies this methodology to two "real world" problems.

II. Experimental Design for Neural Networks

"The basic problem of experimental design is deciding what pattern of design points will reveal aspects of the situation of interest" [8]. Functions met in practice usually show fairly smooth relationships. The simpler the relationship, the fewer the number of experimental points needed to explore it. For example, if it were certain that the relationship between a response η and a single variable x could be represented in a straight line, in the absence of experimental error, only two points would be required to determine it exactly.

Once design points are selected and experiments completed, the results are used to model the system of interest. "Response surface methodology comprises a group of statistical techniques for empirical model building and model exploitation" [19:1]. In classical response surface methodology, model selection often becomes a difficult problem as the analyst attempts to find a model that adequately represents the functional relationship of the variables and the response. In the case of a multilayer perceptron, the form of the model has been decided through the architecture of the network. Therefore, a well-defined method of finding the parameters of the proposed model is already established. In a multilayer perceptron setting, the remaining problem is to select the design points in the feature space that allow us to best estimate these parameters. Box and Hunter assert the need for optimally selecting design points:

> If experiments are not carefully planned, the experimental points may be so situated in the space of the variables that the estimates which can be obtained for the parameter θ are not only imprecise but also highly correlated. Once the data are collected, a statistical analysis, no matter how elaborate, can do nothing to remedy this unfortunate situation. However, by the selection of suitable experimental design in advance, these shortcomings can often be overcome. [12:114]

2.1 Nonlinear Regression

A survey of nonlinear regression is necessary when examining the statistical properties of multilayer perceptrons since they are a specialized form of a nonlinear model. According to White, "backpropagation and nonlinear regression can be viewed as alternative statistical approaches to solving the least squares problem" [68:85]. Neter and Wasserman define nonlinear regression models as "models that are not linear in the parameters and cannot be made so by transformation" [47:550].

Least-squares estimation may be used to determine the parameters in a nonlinear regression model. Suppose there are N observations of $(\mathbf{x}, y)$ where $\mathbf{x}$ is a vector of observations on n variables and y is the univariate response. (Appendix D lists the symbols used.) Then let

$$y_i = f(\mathbf{x}_i; \boldsymbol{\theta}^*) + \varepsilon_i \quad i = 1, \ldots, N \tag{19}$$

where $E[\varepsilon_i] = 0$, $\mathbf{x}_i$ is a $n \times 1$ vector, $\boldsymbol{\theta}^*$ is the $p \times 1$ vector of true (but unknown) parameters, and ε_i are independently and identically distributed (i.i.d) with variance σ^2. The least-squares estimate of $\boldsymbol{\theta}^*$, denoted by $\hat{\boldsymbol{\theta}}$, minimizes the error sum of squares, i.e.,

$$\hat{\boldsymbol{\theta}} = \arg\min[S(\boldsymbol{\theta})] \tag{20}$$

where

$$S(\boldsymbol{\theta}) = \sum_{i=1}^{N} [y_i - f(\mathbf{x}_i; \boldsymbol{\theta})]^2 \tag{21}$$

When each $f(\mathbf{x}_i; \boldsymbol{\theta})$ is differentiable with respect to $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}$ will satisfy

$$\left[\frac{\partial S(\boldsymbol{\theta})}{\partial \theta_t} \right]_{\hat{\boldsymbol{\theta}}} = 0 \qquad t = 1, 2, \ldots, p \tag{22}$$

We will use the following notation:

$$f_i(\boldsymbol{\theta}) = f(\mathbf{x}_i; \boldsymbol{\theta}) \tag{23}$$

$$\mathbf{f}(\boldsymbol{\theta}) = (f_1(\boldsymbol{\theta}), f_2(\boldsymbol{\theta}), \dots, f_N(\boldsymbol{\theta}))^T \tag{24}$$

and the $N \times p$ matrix of first partials is defined as

$$F.(\boldsymbol{\theta}) = \frac{\partial \mathbf{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} = \left\{ \left(\frac{\partial f_i(\boldsymbol{\theta})}{\partial \theta_t} \right) \right\} \tag{25}$$

Using this notation,

$$S(\boldsymbol{\theta}) = [\mathbf{y} - \mathbf{f}(\boldsymbol{\theta})]^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\theta})] \tag{26}$$

Then Equation 22 implies

$$\left[\sum_{i=1}^{N} (y_i - f_i(\boldsymbol{\theta})) \frac{\partial f_i(\boldsymbol{\theta})}{\partial \theta_t} \right]_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = 0 \qquad t = 1, 2, \dots, p \tag{27}$$

These equations are commonly referred to as the *normal equations* for a nonlinear regression model.

Seber and Wild provide the following theorem:

Theorem 1 *Given appropriate regularity conditions, then for large N,*

$$(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*) \sim N_p(0, \sigma^2 C^{-1}) \tag{28}$$

where $C = F.^T F. = F.^T(\boldsymbol{\theta}^*) F.(\boldsymbol{\theta}^*)$ [59:24].

21

It should be noted that the normality of the residuals (ε) is not required for this result. If $\hat{F}. = F.(\hat{\theta})$, then an estimate for C is given by $\hat{C} = \hat{F}.^T \hat{F}.$ An estimate for σ^2 is given by

$$\hat{\sigma}^2 = \frac{S(\hat{\theta})}{N - p} \tag{29}$$

Gallant presents the regularity conditions used to arrive at these results [25:19-21]. In summary, the conditions are:

1. The response function $f(x_i; \theta)$ must be continuous in the argument $(x_i; \theta)$ and the first and second derivatives must be continuous in $(x_i; \theta)$.

2. The sequence of input vectors behave properly as N tends to infinity. Proper behavior is obtained when observations are chosen randomly or are the replication of a fixed set of points.

3. Identification Condition:

$$\lim_{N \to \infty} \frac{1}{N} \sum_{s=1}^{N} [f(x_s; \theta) - f(x_s; \theta^*)]^2 \tag{30}$$

has a unique minimum at $\theta = \theta^*$.

4. Rank Qualification:

$$\lim_{N \to \infty} \frac{1}{N} F.^T(\theta^*) F.(\theta^*) \tag{31}$$

is nonsingular.

In classical nonlinear regression, the estimated parameters $\hat{\theta}$ are determined numerically. A common algorithm is the Gauss-Newton method which makes use of a linear Taylor series expansion. When multilayer perceptrons are used, the parameters θ are estimated using the backpropagation algorithm described in Chapter I. (In that context, the notation for the parameters is w.)

To this point, only single response models have been considered. A natural extension is to replace the single response y_i by an $r \times 1$ vector of responses $\mathbf{y}_i$. This model can be expressed as

$$\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i; \boldsymbol{\theta}) + \boldsymbol{\varepsilon}_i \qquad i = 1, 2, \ldots, N \tag{32}$$

where the $\boldsymbol{\varepsilon}_i$ are assumed to be i.i.d. with mean $\mathbf{0}$ and variance-covariance matrix Σ. In this case, a least-squares estimate of $\boldsymbol{\theta}^*$ can be obtained by minimizing

$$T(\boldsymbol{\theta}) = \sum_{i=1}^{N} [\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i; \boldsymbol{\theta})]^T \Sigma^{-1} [\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i; \boldsymbol{\theta})] \tag{33}$$

with respect to $\boldsymbol{\theta}$ [59:531]. Since the relationship between the responses is typically unknown, some estimate of Σ is usually required in order to estimate the true parameter values $\boldsymbol{\theta}^*$. The elements of Σ can be estimated with the least-squares estimate of $\boldsymbol{\theta}^*$ as:

$$\hat{\sigma}_{uv} = \frac{1}{n} \mathbf{e}_u^T \mathbf{e}_v \qquad u, v = 1, 2, \ldots, r \tag{34}$$

where $\mathbf{e}_j = \mathbf{y}^{(j)} - \mathbf{f}^{(j)}(\hat{\boldsymbol{\theta}})$ (denoting the the jth response model by the superscript j) and $\hat{\Sigma} = \{\hat{\sigma}_{uv}\}$. The problem of estimating Σ to find optimal parameters is not applicable when multilayer perceptrons are used because the backpropagation training method does not consider the variance-covariance matrix.

Define the Kronecker product of A (an $m \times m$ matrix) and B (an $n \times n$ matrix) as:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mm}B \end{bmatrix} \tag{35}$$

Then, an approximation of the variance-covariance matrix of the parameters analogous to $\hat{\sigma}^2(\hat{F}.^T\hat{F}.)^{-1}$ in the single response case is given by the $p \times p$ matrix $\hat{W}^{-1}$ where [59:532]:

$$\hat{W} = \frac{1}{N}F.^T(\hat{\theta})[\hat{\Sigma}^{-1} \otimes I_N]F.(\hat{\theta}) \tag{36}$$

and $F.(\hat{\theta})$ is the $Nr \times p$ matrix of first partials defined as

$$F.(\hat{\theta}) = \frac{\partial \mathbf{f}(\hat{\theta})}{\partial \theta^T}$$

$$= \begin{bmatrix} F._1(\hat{\theta}) \\ F._2(\hat{\theta}) \\ \vdots \\ F._r(\hat{\theta}) \end{bmatrix}$$

where $\mathbf{f}(\theta) = (\mathbf{f}_1(\theta), \mathbf{f}_2(\theta), \ldots, \mathbf{f}_N(\theta))^T$, $\mathbf{f}_i(\theta) = \mathbf{f}(\mathbf{x}_i; \theta)$ and $F._j = \{(\frac{\partial f_j(\mathbf{x}_u; \theta)}{\partial \theta_v})\}$ is the $N \times p$ matrix of derivatives for the jth model.

In the discussion above, $\mathbf{f}(\mathbf{x}_i; \theta)$ is assumed to be the true model. However, in many situations one has no knowledge of the true model and $\mathbf{f}(\mathbf{x}_i; \theta)$ is selected on empirical grounds from a range of possible models [59:572]. For the multilayer perceptrons used in this research, the overall functional form of the model is fixed (two layers of weights and sigmoidal activations) and only the appropriate number of hidden nodes must be chosen.

White describes conditions under which one can form a consistent estimator of C, the variance-covariance matrix of the parameters, even when the model has been misspecified. White calls this estimator *specification robust* [68:259-288]. The practical application of this $\hat{C}$ can have disadvantages [61]. In this research, it will be assumed that a multilayer perceptron with the minimum number of hidden nodes required to achieve a predetermined level of accuracy is an "appropriate" model. Due to the problems associated with determining

a specification robust $\hat{C}$, any inaccuracies due to misspecification of the model will be ignored and "appropriate" multilayer perceptron architectures will be used.

2.2 Experimental Designs in Nonlinear Situations.

Box and Lucas present a method for the design of experiments in nonlinear situations. Suppose that some response η is a known function

$$\eta = f(x_1, x_2, \ldots, x_n; \theta_1, \theta_2, \ldots, \theta_p) = f(\mathbf{x}; \boldsymbol{\theta}) \tag{37}$$

of n variables whose levels are denoted by the elements $x_1, \ldots, x_k, \ldots, x_n$ of the vector x and of p parameters $\theta_1, \ldots, \theta_t, \ldots, \theta_p$ elements of the vector $\boldsymbol{\theta}$, and that this function is not necessarily linear in either the variables or the parameters [13].

The goal here is to select N trials such that they can be expected to provide results from which the p parameters can be estimated with high accuracy. Box and Lucas define the design matrix as an $N \times n$ matrix $\mathbf{D}$. The sth row of this matrix provides the levels of the n variables at which the response is to be observed in the sth trial. In practice, the choice of design points is restricted either by physical or experimental constraints. In general, there will exist a "region of operability," $\mathcal{R}$, in the x-space which defines the area where experiments can be performed. In some cases, the experimental region can be defined by a series of inequalities in the x's such as

$$x_k(\min) \leq x_k \leq x_k(\max) \quad k = 1, 2, \ldots, n \tag{38}$$

$\mathcal{R}$ may, however, be more complicated [13].

Box and Lucas denote the response observed at the sth set of experimental conditions by y_s and suppose that

$$E(y_s) = \eta_s = f(\mathbf{x}_s; \boldsymbol{\theta}) \tag{39}$$

25

and

$$E\left[(y_s - \eta_s)(y_u - \eta_u)\right] = \begin{cases} \sigma^2 & s = u \\ 0 & s \neq u \end{cases} \tag{40}$$

where, $s, u = 1, 2, \ldots, N$ and, in general, σ^2 is unknown. Let the true values of the parameters be denoted by $\theta_1^*, \theta_2^*, \ldots, \theta_p^*$, the elements of the vector $\boldsymbol{\theta}^*$. The partial derivatives of the response function with respect to the tth parameter θ_t for the sth set of experimental conditions, taken at the point $\boldsymbol{\theta}^*$, is denoted by

$$\left[\frac{\partial f(\mathbf{x}_s; \boldsymbol{\theta})}{\partial \theta_t}\right]_{\boldsymbol{\theta} = \boldsymbol{\theta}^*} \qquad s = 1, \ldots, N; t = 1, \ldots, p \tag{41}$$

and the $N \times p$ matrix of these derivatives is $F.(\boldsymbol{\theta})$.

Now, the least squares estimates $\hat{\theta}$ obtained by minimizing the sum of squares

$$\sum_{s=1}^{N} \{y_s - f(\mathbf{x}_s; \boldsymbol{\theta})\}^2 \tag{42}$$

and given by the normal equations

$$\left[\sum_{s=1}^{N} \{y_s - f(\mathbf{x}_s; \hat{\boldsymbol{\theta}})\} \left\{\frac{\partial f(\mathbf{x}_s; \boldsymbol{\theta})}{\partial \theta_t}\right\}\right]_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}} = 0 \qquad t = 1, 2, \ldots, p \tag{43}$$

have a variance-covariance matrix which is approximated by $\sigma^2 (F.^T F.)^{-1}$. Box and Lucas proceed by attempting to choose $\mathbf{D}$ so that the determinant $|(F.^T F.)^{-1}|$ is made as small as possible [13]. That is,

$$\mathbf{D} = \arg \min_{\mathbf{X} \in \mathcal{R}} \left|\left(F.^T(\mathbf{X}; \boldsymbol{\theta}) F.(\mathbf{X}; \boldsymbol{\theta})\right)^{-1}\right| \tag{44}$$

where $\mathbf{X}$ is a set of N data points.

Atkinson and Hunter explain this criterion by examining the boundary of the region with confidence coefficient $1 - \alpha$ in the space of parameters. This boundary is formed by the

values of θ which satisfy the relationship

$$(\theta - \hat{\theta})^T F.^T F.(\theta - \hat{\theta}) = s^2 p F_\alpha(p, \nu) \tag{45}$$

where $F_\alpha(p, \nu)$ is the $\alpha \cdot 100$ percent point of the F-distribution with p and ν degrees of freedom and s^2 is an independent estimate of the error variance σ^2 based on ν degrees of freedom [1].

The boundary of such a region is hyper-ellipsoidal with the volume depending on $|F.^T F.|$ such that the volume will decrease as the value of the determinant increases; hence, the approach recommended by Box and Lucas is to minimize $|(F.^T F.)^{-1}|$ [1]. See Appendix A.1 for an explanation of the proportionality of volume and $|F.^T F.|$. The minimization of $|(F.^T F.)^{-1}|$ is directly related to the minimization of the generalized variance which is defined as the determinant of the variance-covariance matrix. Appendix A.2 shows this relationship. In Box and Lucas's 1959 paper, they give a detailed explanation of the relationship between the sample space, the solution locus within the sample space, and the parameter space. They conclude that in the nonlinear situation, minimization of $|(F.^T F.)^{-1}|$ can be treated as in the linear case depending on the degree of nonlinearity of the function [13]. Notice that the criterion can be simplified for the special case where the number of experiments is equal to the number of parameters. $F.$ is a square $p \times p$ matrix and $|F.^T F.| = |F.|^2$. It is sufficient in this case to maximize $|F.|$.

A *simplex* is defined as the geometrical figure consisting, in N dimensions, of $N + 1$ vertices and all their interconnecting line segments, polygonal faces, etc. In two dimensions, a simplex is a triangle [51]. Atkinson and Hunter state that in the p-dimensional $F.$-space that the value of $|F.|$ is proportional to the volume of the simplex formed by the origin and the p experimental points. Therefore, the goal is also to maximize the volume of this simplex. See Appendix A.3 for further explanation.

The efficiency of the different possible designs depends on the matrix F. with elements equal to the values of the derivatives of the response function with respect to the parameters at $\theta = \theta^*$. The derivatives can only be independent of the values actually taken by the parameters if the response function is linear. For nonlinear response functions the values of the derivatives, and therefore the efficiency of the design, depend on the *actual* values of the parameters. If the goal is to design an effective experiment, then assumptions must be made about the values of the parameters in advance. Here, Box and Lucas assume that preliminary values of the parameters $\hat{\boldsymbol{\theta}}$ are available and proceed as if these were the true quantities [13].

A simple example illustrates the method. Let

$$\eta = \frac{\theta_1}{\theta_1 - \theta_2} \left[\exp(-\theta_2 x_1) - \exp(-\theta_1 x_1) \right] \tag{46}$$

The problem is to choose a set of values $x_{s1}, s = 1, 2, \ldots, N$, at which to observe η so that from these observations θ_1 and θ_2 can be estimated as accurately as possible (allowing for experimental error in the observations). In this example, the design matrix $\mathbf{D}$ would consist of a single column whose N entries are the values $x_{11}, \ldots, x_{s1}, \ldots, x_{N1}$ at which η is to be observed and $\boldsymbol{\theta}$ contains the two elements θ_1 and θ_2.

Suppose two design points x_{11} and x_{21} are required. Given preliminary guesses $\hat{\theta}_1, \hat{\theta}_2$, which values of x_{11} and x_{21} should be chosen so that the best estimates will be available for θ_1 and θ_2? The values x_{11} and x_{21} should be chosen to maximize the determinant $|\hat{F}.^T \hat{F}.|$. In this case, since the number of trials is equal to the number of parameters

$$|\hat{F}.^T \hat{F}.| = |\hat{F}.|^2 \tag{47}$$

and minimization of $|\hat{F}.|^2$ is equivalent to the minimization of $|\hat{F}.|$.

$$|\hat{F}.| = \begin{vmatrix} \dfrac{\partial f(x_{11};\boldsymbol{\theta})}{\partial \theta_1} & \dfrac{\partial f(x_{11};\boldsymbol{\theta})}{\partial \theta_2} \\[2mm] \dfrac{\partial f(x_{21};\boldsymbol{\theta})}{\partial \theta_1} & \dfrac{\partial f(x_{21};\boldsymbol{\theta})}{\partial \theta_2} \end{vmatrix} \tag{48}$$

If the preliminary guesses are

$$\hat{\theta}_1 = 0.7, \quad \hat{\theta}_2 = 0.2 \tag{49}$$

then the values of $x_{11} = 1.23$ and $x_{21} = 6.86$ maximize the determinant and these are the desired design points.

Although an analytic solution was possible for this example, numerical methods must be employed as the size and complexity of the problem increases. Well-defined nonlinear optimization methods are numerous. Reklaitis *et al.* list several direct search methods such as the simplex search method, the Hooke-Jeeves pattern search method and Powell's conjugate direction method. Gradient-based methods are also available, such as Newton's method, quasi-Newton methods and the conjugate gradient method [54].

Atkinson and Hunter extend the method of Box and Lucas to include cases where N, the number of experiments, is greater than p, the number of parameters. They also establish conditions under which replications of p experiments form an optimal design for N experiments when N is a multiple of p [1].

2.3   Sequential Designs in Nonlinear Situations

The method described above determines the design points as a group, before the experiments are conducted. Given that the observations have been made at N_0 design points, a sequential strategy chooses the next N points in some optimal fashion. This strategy may be superior to the "all-at-once" approach since the best selection of the experimental conditions

29

for nonlinear models depends on the values of the parameters themselves. The theory for non-sequential designs may be established by setting $N_0 = 0$.

A sequential method assumes the results of N_0 experiments are available in planning the $(N_0 + N)$th experiment. Box and Hunter discuss how to add points one at a time by maximizing the peak of the posterior distribution of θ based on $N_0 + N$ points [12]. Using a uniform prior for θ and various approximations, they showed that this criterion leads to maximizing $|F.^T(\theta)F.(\theta)|$ as before, but with respect to just $x_{N_0+1}, \ldots, x_{N_0+N}$. Their results are extended to the case where θ has a multivariate normal prior by Draper and Hunter [21].

For the sake of illustration, assume $N = 1$, i.e., additional points are introduced one at a time. Let $f._{(N_0+1)} = \frac{\partial f(x_{N_0+1};\theta)^T}{\partial \theta}$. Then for $N_0 + 1$ observations, $|C_{(N_0+1)}|$ is to be maximized where

$$
\begin{aligned}
|C_{(N_0+1)}| &= \left| \begin{pmatrix} F.(\theta) \\ f._{(N_0+1)}^T \end{pmatrix}^T \begin{pmatrix} F.(\theta) \\ f._{(N_0+1)}^T \end{pmatrix} \right| \qquad (50)\\
&= |F.^T(\theta)F.(\theta) + f._{(N_0+1)}f._{(N_0+1)}^T| \\
&= |C_{(N_0)} + f._{(N_0+1)}f._{(N_0+1)}^T| \\
&= |C_{(N_0)}|(1 + f._{(N_0+1)}^T C_{(N_0)}^{-1} f._{(N_0+1)}) \qquad (51)
\end{aligned}
$$

See Appendix B for details on this development. Since $|C_{(N_0)}|$ does not involve x_{N_0+1}, the above criterion reduces to maximizing $f._{(N_0+1)}^T C_{(N_0)}^{-1} f._{(N_0+1)}$ with respect to x_{N_0+1}.

The parameter vector can be estimated by $\hat{\theta}_{(N_0)}$ after N_0 observations and updated to $\hat{\theta}_{(N_0+1)}$ after observation $N_0 + 1$. It can be shown that when $\hat{\theta}_{(N_0)}$ is close to θ^*, the design criteria is also equivalent to finding x_{N_0+1} to maximize $f._{(N_0+1)}^T C_{(N_0)}^{-1} f._{(N_0+1)}$ which may be interpreted as maximizing the asymptotic variance of the prediction $f(x_{N_0+1}; \hat{\theta}_{(N_0+1)})$ [59:258].

Although these sequential methods may be superior in some nonlinear regression settings, there may be problems when applying the methods to multilayer perceptrons. The principal problem is that parameters must be re-estimated after each set of N observations. If N is small, repeated re-estimation will be required and the multilayer perceptron training may be excessive.

2.4 Design of Experiments in Multi-response Situations.

To draw inferences about parameters in multi-response estimation, one often uses a Bayesian formulation. Box and Draper [10] establish Bayesian regions for parameters in multi-response situations. These regions, which Box and Tiao called highest posterior density (h.p.d.) regions, play a role parallel to that played by confidence regions in sampling theory analysis. To investigate these regions, define the quantities

$$v_{ij} = \sum_{s=1}^{N_0} [y_{is} - f_i(\mathbf{x}_s; \boldsymbol{\theta})] [y_{js} - f_j(\mathbf{x}_s; \boldsymbol{\theta})] \qquad i, j = 1, \dots, r \tag{52}$$

and suppose N_0 sets of observations

$$\mathbf{y}_s^T = (y_{1s}, y_{2s}, \dots y_{rs}) \qquad s = 1, 2, \dots, N_0 \tag{53}$$

have already been obtained. Then, since the N_0 set of observations are independent, the likelihood is

$$p(\mathbf{y}|\boldsymbol{\theta}, \sigma^{ij}) = (2\pi)^{-\frac{1}{2}N_0 r} |A|^{\frac{1}{2}N_0} \exp\left\{ -\frac{1}{2} \sum_{i=1}^{r} \sum_{j=1}^{r} \sigma^{ij} v_{ij} \right\} \tag{54}$$

where $A = \Sigma^{-1} = \{\sigma^{ij}\} = \{\sigma_{ij}\}^{-1}$ [10].

Since little is known *a priori* about the values of the parameters, a locally uniform prior distribution is used [10:356] so that

$$p(\boldsymbol{\theta})d\boldsymbol{\theta} \propto d\boldsymbol{\theta} \tag{55}$$

Then the posterior distribution for $\boldsymbol{\theta}$ after N_0 observations is proportional to

$$p_{N_0}(\boldsymbol{\theta}|\sigma^{ij},\mathbf{y})d\boldsymbol{\theta} \tag{56}$$

According to Draper and Hunter, this posterior distribution can be used as the prior distribution to determine the posterior distribution after N further observations are made. The new posterior distribution for $\boldsymbol{\theta}$ after $N_0 + N$ observations is proportional to

$$p_{N_0+N}(\boldsymbol{\theta}|\sigma^{ij},\mathbf{y})d\boldsymbol{\theta} = (2\pi)^{-\frac{1}{2}(N_0+N)r}|A|^{\frac{1}{2}(N_0+N)}\exp\left\{-\frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{r}\sigma^{ij}v_{ij}\right\} \tag{57}$$

with the upper summation limits of v_{ij} now extending to $N_0 + N$.

The goal, then, is to select the N observations in such a way that the posterior density (Equation 57) obtained after $N_0 + N$ observations is maximized with respect to $\boldsymbol{\theta}$ and with respect to the N observations to be chosen. Of course, from a Bayesian point of view, all the information concerning the parameters $\boldsymbol{\theta}$ is contained in the posterior distribution *after* the observations are obtained. Therefore, the goal is altered to achieve the best possible posterior distribution by proper choice of design *before* the observations have actually been obtained [20:528].

Assume that for a region in the $\boldsymbol{\theta}$-space sufficiently close to the maximum likelihood estimates $\hat{\boldsymbol{\theta}}$ that the following holds:

$$f_i(\mathbf{x}_s;\boldsymbol{\theta}) = f_i(\mathbf{x}_s;\hat{\boldsymbol{\theta}}) + \sum_{t=1}^{p}(\theta_t - \hat{\theta}_t)f._{is}^{(t)} \qquad i = 1,\ldots,r; s = 1,\ldots,N \tag{58}$$

where

$$f_{\cdot is}^{(t)} = \left[\frac{\partial f_i(\mathbf{x}_s; \boldsymbol{\theta})}{\partial \theta_t} \right]_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}} \tag{59}$$

Letting,

$$\delta_{is} = y_{is} - f_i(\mathbf{x}_s; \hat{\boldsymbol{\theta}}) \tag{60}$$

it follows that one can write, correct to second order in $\boldsymbol{\theta}$,

$$\sum_{i=1}^{r} \sum_{j=1}^{r} \sigma^{ij} v_{ij} = \sum_{i=1}^{r} \sum_{j=1}^{r} \sigma^{ij} \sum_{s=1}^{N} \delta_{is} \delta_{js} + \sum_{i=1}^{r} \sum_{j=1}^{r} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T \left\{ \sigma^{ij} F_{\cdot i}^T F_{\cdot j} \right\} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \tag{61}$$

There are no terms linear in $(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$ due to the definition of $\hat{\boldsymbol{\theta}}$ as the maximum likelihood estimate.

Draper and Hunter state in their development that after performing the appropriate normalization and using Equation 61 in Equation 57,

$$p_{N_0+N}(\boldsymbol{\theta} | \sigma^{ij}, \mathbf{y}) = (2\pi)^{-\frac{1}{2}P} |D|^{\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T D (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \right\} \tag{62}$$

where

$$D = \sum_{i=1}^{r} \sum_{j=1}^{r} \sigma^{ij} F_{\cdot i}^T F_{\cdot j} \tag{63}$$

By definition, maximization with respect to $\boldsymbol{\theta}$ occurs when $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$, evaluated after $N_0 + N$ observations. It thus remains to choose the design points so that the determinant

$$|D| = \left| \sum_{i=1}^{r} \sum_{j=1}^{r} \sigma^{ij} F_{\cdot i}^T F_{\cdot j} \right| \tag{64}$$

is maximized in the case where the σ^{ij} are known [20].

M. J. Box develops a similar criterion for the case when one is interested in only a subset of the parameters with Σ known, but possibly non-constant [14]. Wijesinha and Khuri

extended these results by presenting the sequential construction of optimal designs when the variance-covariance matrix of the responses Σ is unknown [69].

M.J. Box and Draper [15] rely on results from Box and Draper [10] and Draper and Hunter [20] to derive a design point criterion for non-sequential multi-response design of experiments when Σ is unknown. Box and Draper derive the distribution of θ and give

$$p(\boldsymbol{\theta}|\mathbf{y}) = \mathcal{C}|v_{ij}|^{-\frac{1}{2}N} \tag{65}$$

where

$$\mathcal{C} = \left[\int |v_{ij}|^{-\frac{1}{2}N} d\boldsymbol{\theta} \right]^{-1} \tag{66}$$

is the normalizing constant. Parameter estimation can be accomplished by minimizing $|v_{ij}|^{-\frac{1}{2}N}$. This criterion for parameter estimation chooses

> ... that value $\hat{\theta}$ of θ for which the posterior density is a maximum. This maximum posterior density is a function of the experimental settings adopted, so that if some or all of the experimental settings have not yet been selected, they can be chosen so as to maximize the maximum of the posterior density with respect to θ. [15:17-18]

In order to get a useful design criterion, second order approximations to the posterior density are used. M.J. Box and Draper estimate $p(\boldsymbol{\theta}|\mathbf{y})$ correct to second order in $(\theta - \hat{\theta})$ by

$$p^*(\boldsymbol{\theta}|\mathbf{y}) = \mathcal{C}k \exp \left\{ -\frac{1}{2}(\theta - \hat{\theta})^T A(\theta - \hat{\theta}) \right\} \tag{67}$$

where k is a constant and $A = \sum_{i=1}^{r} \sum_{j=1}^{r} \hat{v}^{ij} F_{\cdot i}^T F_{\cdot j}$ with

$$\hat{v}_{ij} = \frac{1}{N} \sum_{s=1}^{N} \left[y_{is} - f_i(\mathbf{x}_s; \hat{\boldsymbol{\theta}}) \right] \left[y_{js} - f_j(\mathbf{x}_s; \hat{\boldsymbol{\theta}}) \right] \tag{68}$$

and $\{\hat{v}^{ij}\} = \{\hat{v}_{ij}\}^{-1}$ for data sets with unknown variance-covariance matrices [15].

As before, it remains to choose design points so that the determinant

$$|\hat{D}| = \left| \sum_{i=1}^{r} \sum_{j=1}^{r} \hat{v}^{ij} F_{\cdot i}^{T} F_{\cdot j} \right| \qquad (69)$$

is maximized. According to Seber and Wild, $\hat{v}^{ij}$ is the maximum-likelihood estimator of Σ^{-1} [59:582]. Further, it is intuitively appealing that the unknown variance-covariance elements are estimated by their natural estimators—$\frac{1}{N}$ times the sums of squares and cross products of the difference between observed and modeled responses [15]. Since, in the design stage, the residuals of the unperformed experiments are unknown, all N_0 data vectors from completed experiments should be used to estimate the variance-covariance elements.

2.5   Discrimination Between Specified Models

To this point, the goal has been to choose design points to best estimate the parameters of the chosen model. If, instead, the goal is to choose design points to best discriminate between two models, different techniques are required. Box and Hill developed an approach that supposes that N_0 observations have already been taken (at least enough to initially estimate the parameters) and considers where best to put the $(N_0 + 1)$st for maximum discrimination between the models. The $(N_0 + 1)$st observation is chosen at levels which will maximize the expected decrease in entropy from the N_0th to the $(N_0 + 1)$st experiment [11].

Box and Hill's method extends to more than two models. They demonstrate an example in which the simplest model is the truth model and all other models are generalizations of the simple model containing more parameters. One would expect the entropy measure to have difficulty choosing between the models. However, their criterion tended toward selection of the simplest model [11]. These results indicate that designing an experiment with correctly chosen points may be a way to choose a parsimonious nonlinear model.

2.6 Classical Optimality Criteria

The previous sections outlined design of experiments methods in which the determinant of the estimated variance-covariance matrix is minimized (called a D-optimal criterion). According to Mitchell [44], D-optimality is good in many respects

1. Low variance for the parameters

2. Low correlations among parameters

3. Low maximum variance of estimated responses

The D-criterion, as discussed in previous sections is equivalent to minimizing the determinant of the asymptotic estimate of the variance-covariance matrix. In addition, the determinant is proportional to the volume of the asymptotic confidence ellipsoid.

There are other possible measures. According to Pukelsheim, "the ultimate purpose of any optimality criterion is to measure 'largeness' of a nonnegative definite $s \times s$ matrix C" [52:135]. Table 2 shows the most often used measures. The D-criterion is a measure of region size alone, whereas the A-criterion and the E-criterion measure size *and* sphericity [9:491]. The A-criterion is especially appealing if the parameters have definite physical meaning [52:137]. The E-criterion minimizes the maximum variance and requires a method of determining eigenvalues. The final criterion—the trace—is by itself "rather meaningless" [52:138]. Yet, T-optimality can be useful if accompanied by further conditions. It appears in the literature that this criteria has been used almost exclusively for linear designs. One important result in design theory is the general equivalence theorem which links D- and G-optimality [36].

Table 2. Optimality Criteria

D-Optimality	Determinant Criterion	$D = |(F.^T F.)^{-1}|$
A-Optimality	Average-Variance Criterion	$A = \text{tr}[(F.^T F.)^{-1}]$
E-Optimality	Smallest-Eigenvalue Criterion	$E = \max \lambda_i$ λ_i are the eigenvalues of $(F.^T F.)^{-1}$
T-Optimality	Trace Criterion	$T = [\text{tr}(F.^T F.)]^{-1}$
G-Optimality	Prediction Variance Criterion	$G = \max_{x \in R}[f.^T(x)(F.^T F.)^{-1} f.(x)]$

2.7 Choosing Training Vectors for Multilayer Perceptrons

The literature reviewed in this chapter so far has been from a large body of "classical" nonlinear regression sources. Published research on designing experiments specifically for multilayer perceptrons is much sparser. Several related topics are reviewed in this section.

MacKay derives a criterion that measures how useful a data point is expected to be. He suggests using the criterion as a guide to selecting points for what he calls "active learning." His strategy centers around maximizing the expected change in mean marginal entropy of a distribution over w for points in the feature space [41].

Baum embeds the idea of queries in a neural network training algorithm. In a query, the algorithm supplies an exemplar and is told the classification of the vector by an oracle. Using examples from a training set, the algorithm determines where to query in order to gain information on separating hyperplanes. Baum claims that his algorithm is quite efficient in the number of queries that it uses and in the amount of time required to train [3].

Hwang *et al.* propose a query-based approach that samples on and near the current decision boundaries. Boundaries are generated by a network inversion algorithm and gradients are calculated for boundary points. Using the gradient information, conjugate pairs are generated. A pair consists of two points lying on opposite sides of the line passing through

the boundary point and perpendicular to the boundary surface, with their distances to the corresponding boundary point equal to $1/|\text{gradient}|$ [34].

MacKay argues that the strategies of Hwang and Baum are "human-designed strategies and it is not clear what objective function if any they optimize" [41:728]. Further, he criticizes Hwang's method for the following reasons:

- "If we have already sampled a great deal on one particular boundary then we do not gain useful information by repeatedly sampling there either, because the location of the boundary has already been established!"

- "A strategy that samples only near existing boundaries is not likely to make new discoveries."

- "To be efficient, a strategy should take into account how influential a datum will be" [41].

Atlas *et al.* propose a method of selectively sampling data from regions in the domain that are unknown based on information from previous batches of samples. The novelty in their approach is that the points selected to train on are determined using two multilayer perceptrons in parallel. Both networks are trained with known examples and with random "background" patterns. One network is trained to classify the background patterns as positive and the other is trained to classify the background patterns as negative. The region of uncertainty is then "captured" by taking the symmetric difference of the outputs of the two networks [2].

None of the works outlined in this section employ the statistically based D-optimality criterion. The reason for this omission may stem from the large number of parameters used in the multilayer perceptron. Another reason may be that in the past neural network practitioners were rarely involved in the collection of training data. However, as the field matures and

experimenters themselves use the networks, choosing optimal design points will become increasingly important.

2.8   Chapter Summary

This chapter has introduced experimental design methods. The procedure developed by Box and Lucas is key to the research presented in this document. The central idea behind their procedure is the minimization of a criterion based on the volume of the confidence ellipsoid. The extension of the univariate criterion to a multivariate criterion is also an important concept which will be explored in Chapter IV. Finally, the equation obtained for augmenting an existing design (Equation 51) will be used extensively in subsequent chapters.

The next chapter puts into practice the univariate methods and tailors them specifically for multilayer perceptrons. A procedure for ranking design points is developed and simplifications to the basic method are introduced.

III. Design of Experiments for Single Output Multilayer Perceptrons

This chapter investigates the design of experiments for multilayer perceptrons with a single output. The research and theoretical results are based on the use of a multilayer perceptron with a single layer of middle nodes and sigmoidal activations at the middle and output nodes. Additionally, backpropagation was used to obtain weights (train) for all multilayer perceptrons used. Chapter I provides details.

Two-class discrimination problems require a single output in a multilayer perceptron. A vector is classified as Class 1 if the output is less than 0.5 and as Class 2 if the output is greater than 0.5. Networks with a single output correspond to nonlinear regression models with a single output. It is in this arena that much of the literature on the design of experiments is centered.

In this chapter, the Box and Lucas method for obtaining D-optimal designs is explored further. Next, maximization routines which are essential to the design method are explored. Results are then presented on both a "simple" linearly separable discrimination problem and a more complex nonlinearly separable discrimination problem. The development and testing of a ranking scheme for a set of design points is given next. Attempts to reduce the complexity of the determination of design points are introduced, and finally the sensitivity of the method is examined.

3.1 Introduction

3.1.1 A Further Look at D-Optimality. To look at the effect that maximizing the determinant $|F^T F|$ has on the choice of design points, consider the case in which there are just two parameters. Let

$$C = F^T F. \tag{70}$$

$$= \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

and

$$C^{-1} = \begin{bmatrix} c^{11} & c^{12} \\ c^{21} & c^{22} \end{bmatrix} \qquad (71)$$

$$= \frac{1}{|C|} \begin{bmatrix} c_{22} & -c_{12} \\ -c_{21} & c_{11} \end{bmatrix}$$

The terms c^{ii} are proportional to the variances of the estimated parameters $\hat{\theta}_i$ and $c^{ij}(i \neq j)$ proportional to the covariances. The criterion is to minimize the absolute value of $|C^{-1}|$. For the two parameter case

$$|C^{-1}| = c^{11}c^{22} - 2c^{12} \qquad (72)$$

Considering the case where there is no correlation between the estimates, then it is desirable to make c^{22} and c^{11} as small as possible. This corresponds to choosing design points to minimize the variance of $\hat{\theta}_1$ and $\hat{\theta}_2$.

If a correlation between $\hat{\theta}_1$ and $\hat{\theta}_2$ exists, then since C^{-1} is positive definite, $|C^{-1}|$ is positive and $c^{11}c^{22} - 2c^{12} > 0$ and $c^{11}c^{22} > 2c^{12}$. Therefore, minimizing $c^{11}c^{22}$ infers minimizing $c^{12} = c^{21}$ which are the parameter covariances. In summary, the criterion being used will tend to pick out a set of design points **D** which will minimize the variance of the individual parameters *and* minimize the covariance between the parameters.

3.1.2 Notation. To facilitate a multilayer perceptron specific application, further notation is required. Typically, the parameters in a multilayer perceptron are called *weights* and the set of all weights denoted **w**. Since there is no natural ordering for these weights, an ordering will be established here. The weights below the hidden layer will be listed first with

all the weights emanating from an input node listed together. Then the weights above the hidden layer will be listed with the weights emanating from a hidden node listed together. So,

$$\mathbf{w} = (w_{11}^1, w_{12}^1, \ldots, w_{1m}^1, w_{21}^1, w_{22}^1, \ldots, w_{2m}^1, \ldots, w_{nm}^1, \xi_1^1, \xi_2^1,$$

$$\ldots, \xi_m^1, w_{11}^2, w_{12}^2, \ldots, w_{1r}^2, w_{21}^2, w_{22}^2, \ldots, w_{2r}^2, \ldots, w_{mr}^2, \xi_1^2, \xi_2^2, \ldots, \xi_r^2)^T \quad (73)$$

where n is the number of inputs (dimension of the input vector), m is the number of middle nodes, r is the number of outputs (in this chapter $r = 1$), and ξ_j^l are the weights connected to the bias elements for the lth layer. The dimension of this weight vector is given by

$$p = (n+1)m + (m+1)r \quad (74)$$

The input vectors will continue to be denoted $\mathbf{x}_s, s = 1, \ldots, N$. The nonlinear regression responses are called *outputs* in the multilayer perceptron arena and will be denoted $\mathbf{z}(\mathbf{x}_s; \mathbf{w}) = (z_1^s, \ldots, z_j^s, \ldots, z_r^s)$ where z_j^s is the jth output node's value for the sth input vector. For the single output multilayer perceptron, the subscript j will be replaced by 1.

The $N \times p$ matrix of first partials becomes

$$F.(\mathbf{w}) = \left\{ \frac{\partial z_1^s}{\partial w_t} \right\} \qquad s = 1, \ldots, N; t = 1, \ldots, p \quad (75)$$

where

$$w_t = \begin{cases} w_{\lceil \frac{t}{m} \rceil, t-m(\lceil \frac{t}{m} \rceil - 1)}^1 & \text{for } t \leq (n+1)m \\ w_{\lceil \frac{t}{(n+1)m+1} \rceil, t-1(\lceil \frac{t}{(n+1)m+1} \rceil - 1) - (n+1)m}^2 & \text{for } (n+1)m < t \leq (n+1)m + (m+1) \end{cases} \quad (76)$$

and $\lceil \alpha \rceil$ is the smallest integer greater than α. Then for lower layer weights,

$$\frac{\partial z_1^s}{\partial w_{ki}^1} = z_1^s(1 - z_1^s)w_{ij}^2 x_i^{1s}(1 - x_i^{1s})x_k^s \quad (77)$$

42

where x_i^{1s} is the activation of the ith middle node given the input vector s and x_k^s is the kth element of the input vector s. Similarly for upper layer weights,

$$\frac{\partial z_1^s}{\partial w_{i1}^2} = z_1^s(1 - z_1^s)x_i^{1s} \tag{78}$$

Note that in order to calculate $F.(\mathbf{w})$, $\mathbf{w}$ (or some estimate of it) must be known. An initial estimate $\hat{\mathbf{w}}$ is obtained by training a multilayer perceptron on existing data vectors. Once $\hat{\mathbf{w}}$ is determined, then $F.(\hat{\mathbf{w}}) = \hat{F}.$ can be formed. A second function of this initial data set is to determine the appropriate network architecture. Steppe develops methods for feature and model selection which can be employed [61].

Using multilayer perceptron notation, the calculation of $\hat{F}.$ has been established. What remains is the maximization of $|\hat{F}.|$, which will be covered in the next section.

3.2   Methods of Maximization

The criterion for determining experimental design points given above requires maximization of the determinant $|\hat{F}.|$ or $|\hat{F}.^T\hat{F}.|$. Methods for maximizing this determinant vary depending on whether the feature space is continuous or discrete. A continuous feature space allows for the selection of any n-dimensional vector in the region of operability ($\mathcal{R}$) as a design point. In contrast, a discrete feature space allows only for the selection of n-dimensional vectors within some set of feasible points.

3.2.1   Continuous Feature Space—Powell's Method.   Powell's maximization method is employed in this research for finding the optimal design points in the case of a continuous feature space [50, 54, 51, 65]. Powell's method is a zero-order method meaning that only evaluation of the original function is used to determine the maximum. The gradient information

43

necessary for higher-order methods is available but, it is in a very complicated form. It is assumed in this research that no gradient information is available.

According to Vanderplaats, Powell's method (including its subsequent modifications) is one of the most efficient and reliable of the zero-order methods [65]. Powell's method is based on the concept of conjugate directions. Given an $n \times n$ symmetric matrix H, the directions $S^{(1)}, S^{(2)}, \ldots, S^{(r)}, r \leq n$ are said to be H conjugate if the directions are linearly independent and [54:92]

$$(S^i)^T H S^j = 0 \quad i \neq j \tag{79}$$

The significance of conjugacy is that given a quadratic function, the function will be maximized in n or fewer conjugate search directions.

Powell's method begins by searching in the n coordinate directions, $S^{(i)}, i = 1, \ldots, n$ where each search updates the location vector X. Having completed the n unidirectional searches, a new search direction is created by connecting the first and last design points. This $(n + 1)$st search direction is conjugate to the previous n directions. The search information is typically stored in a matrix H (The matrix H is chosen by convention because the matrix approximates the Hessian matrix). H begins as an identity matrix. The columns of H represent the unidirectional search vectors $S^{(i)}$. After finding the maximum in any direction, the $S^{(i)}$ in matrix H is replaced by $\alpha_i S^{(i)}$ where the maximum is obtained. The conjugate direction is created as

$$S^{(n+1)} = \sum_{i=1}^{n} \alpha_i S^{(i)} \tag{80}$$

which is the sum of the columns of H. This direction is searched to find α_{n+1}. Each column of H is shifted once to the left and $\alpha_{n+1} S^{(n+1)}$ is stored in column n. This provides a new H matrix containing n directions to start the entire search process over [65]. The search continues until the function effectively stops increasing.

Powell's method breaks down in two situations:

- If some search direction gains no improvement

- If after a few iterations the search directions become parallel either due to roundoff errors or because the function is non-quadratic

According to Vanderplaats, the simplest and most effective way to deal with these problems is to restart the process with the coordinate directions whenever the process slows down[65:86].

Reklaitis *et al.* state that if the function to be maximized is quadratic and has a maximum, then this maximum will be reached in exactly n loops. If the function is not quadratic, then more loops are required. They go on to say that if the function is not quadratic, the method will "converge to a local minimum [maximum] and will do this at a superlinear rate" [54:96].

To implement Powell's method, a convergence check and a test to ensure linear independence must be inserted. As implemented in *Numerical Recipes* [51], the code for Powell's algorithm includes calls to a subroutine which brackets the maximum, a subroutine which performs the line maximization in each of the search directions, and a subroutine which performs the function evaluation.

As described above, Powell's method is an unconstrained maximization algorithm. Maximization of $|\hat{F}.|$ requires maximization within some region of operability, $\mathcal{R}$, as described in Chapter II. It is assumed that all data sets are normalized so that each feature input is in the range [0,1]. The Powell algorithm was modified so that the design points chosen to maximize $|\hat{F}.|$ were in this range. Due to the randomness inherent in this algorithm, multiple runs were accomplished and the run producing the maximum value of the determinant is used.

3.2.2 Discrete Feature Space—Discrete Exchange Algorithm. Nearly all discrete algorithms are based upon "the principles of optimal augmentation and/or reduction of an existing design" [46]. Mitchell's method for maximizing $|\hat{F}.^{T}\hat{F}.|$ based specifically on the

construction of D-optimal experimental designs is one such algorithm [44]. He assumes a linear model. However, with the linearization of a given nonlinear model, the same development applies. Mitchell's algorithm—DETMAX—exchanges design points in the following way:

Starting with a randomly chosen N run design, the initial set of N runs is improved by

1. adding an $(N + 1)$st run, chosen for the maximum possible increase in $|\hat{F}.^T \hat{F}.|$

2. removing that run which results in the minimum possible decrease in $|\hat{F}.^T \hat{F}.|$

What remains to complete the algorithm is the establishment of some criterion to judge the change in $|\hat{F}.^T \hat{F}.|$ when a single design point is added or removed. Mitchell cites Dykstra as having developed the theory for augmenting experimental data to maximize the required determinant and presents the following theorem [44]

Theorem 2 *Let* $\mathbf{X}$ *be the "matrix of independent variables" corresponding to an initial design. If a run at the point* $\mathbf{x}_a$ *is added to the initial design, the new matrix of independent variables is*

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \mathbf{x}_a \end{bmatrix} \tag{81}$$

The following relationship between $|\tilde{\mathbf{X}}^T \tilde{\mathbf{X}}|$ *and* $|\mathbf{X}^T \mathbf{X}|$ *can be shown:*

$$|\tilde{\mathbf{X}}^T \tilde{\mathbf{X}}| = |\mathbf{X}^T \mathbf{X}|(1 + \mathbf{x}_a^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_a) \tag{82}$$

In the case of a linearized nonlinear model, the result becomes

$$|\tilde{F}.^T \tilde{F}.| = |\hat{F}.^T \hat{F}.|(1 + \hat{\mathbf{f}}.^T (\mathbf{x}_a)(\hat{F}.^T \hat{F}.)^{-1} \hat{\mathbf{f}}.(\mathbf{x}_a)) \tag{83}$$

46

with

$$\tilde{F}. = \begin{bmatrix} \hat{F}. \\ \hat{\mathbf{f}}.(\mathbf{x}_a) \end{bmatrix} \tag{84}$$

Then, to make $|\tilde{F}.^T \tilde{F}.|$ as large as possible given the current design, one must choose $\mathbf{x}_a$ so that $\hat{\mathbf{f}}.^T(\mathbf{x}_a)(\hat{F}.^T \hat{F}.)^{-1}\hat{\mathbf{f}}.(\mathbf{x}_a)$ is as large as possible. Similarly, given an $N + 1$ run design, one chooses an $\mathbf{x}_a$ for removal by making $\hat{\mathbf{f}}.^T(\mathbf{x}_a)(\hat{F}.^T \hat{F}.)^{-1}\hat{\mathbf{f}}.(\mathbf{x}_a)$ as small as possible.

In summary, the algorithm begins with a random N run design and adds the vector from the feasible set with the largest value of $\hat{\mathbf{f}}.^T(\mathbf{x}_a)(\hat{F}.^T \hat{F}.)^{-1}\hat{\mathbf{f}}.(\mathbf{x}_a)$ resulting in an $N + 1$ run design. Then, the value of $(\hat{F}.^T \hat{F}.)^{-1}$ is recalculated for the $N + 1$ run design and the vector from the design with the smallest value of $\hat{\mathbf{f}}.^T(\mathbf{x}_a)(\hat{F}.^T \hat{F}.)^{-1}\hat{\mathbf{f}}.(\mathbf{x}_a)$ is removed. If one considered all possible subsets of design points of size N that could be formed from K feasible exemplars,

$$\frac{K!}{N!(K - N)!} \tag{85}$$

subsets would be examined. Applying the discrete criterion will significantly reduce this number. Section 3.3.5 discusses the number of iterations required for a sample problem.

Mitchell's original code for this algorithm was not available, so the algorithm was coded using descriptions from his article "An Algorithm for the Construction of D-Optimal Experimental Designs" [44].

3.3 Results

3.3.1 Linearly Separable Continuous Feature Space. In order to validate the applicability of this approach, a small problem was needed. Figure 6 shows the true linear separator (the "truth model") and the training data used. Two dimensional inputs were used so that the effects of the design method could be shown graphically. A multilayer perceptron with a single hidden node was trained and the weights were recorded. These weights represent the
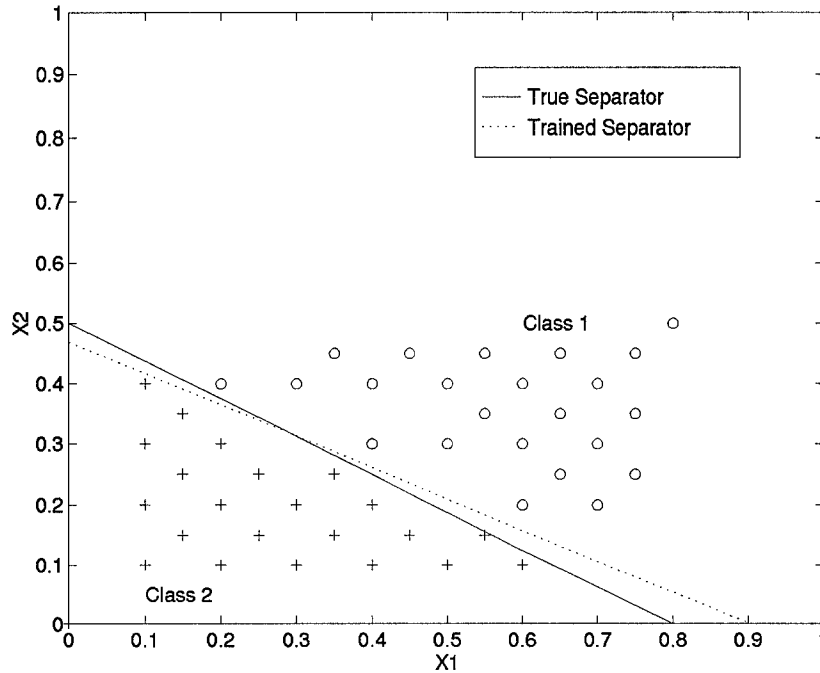
Figure 6. Linearly Separable Classification Problem

initial weight vector ($\hat{w}$) that is required to determine optimal design points. Also shown in Figure 6 is the decision surface for the trained network.

A multilayer perceptron with a single hidden node and two inputs uses five weights. Hence, best exemplars at which to run five future experiments were chosen. It is assumed that all data has been normalized to values between 0 and 1 so that the search for design points will be constrained to values between 0 and 1, i.e., $0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1$. Powell's algorithm was used to maximize the required determinant and obtain the points at which the five future experiments should be conducted. Figure 7 shows the new design points. Since the truth model is known, the design points are simply classified according to this model. The multilayer perceptron was trained again with these five exemplars added to the training set. The resulting boundary is also shown in Figure 7.
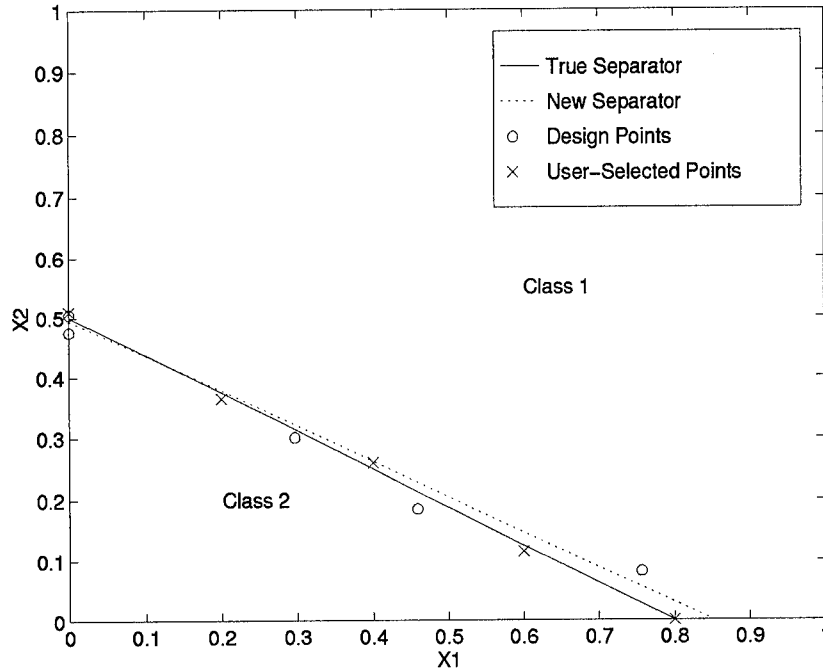
Figure 7. Linearly Separable Problem—Optimal Design Points

For purposes of comparison, the original multilayer perceptron was also trained with an additional five points chosen randomly. Since for this problem the decision surfaces can be seen, five points were chosen near the class boundaries to represent possible "user selected" experiments. These user-selected points were chosen at equal intervals on the x_1 axis near the true separator. The resulting output error for each of these training sessions is shown in Figure 8. The lines in this figure represent the output error at each epoch averaged over 30 training runs. Output error is used here vice classification error due to the simplicity of the classification problem. Since all of the methods demonstrated nearly perfect classification accuracy, differences in performance could only be discerned by using output error.

3.3.2 Research Methodology. For subsequent examples, the design point method will be compared to using randomly selected points and points arranged in a grid. Figure 9 illustrates the research procedure. First, a multilayer perceptron is trained with an initial data
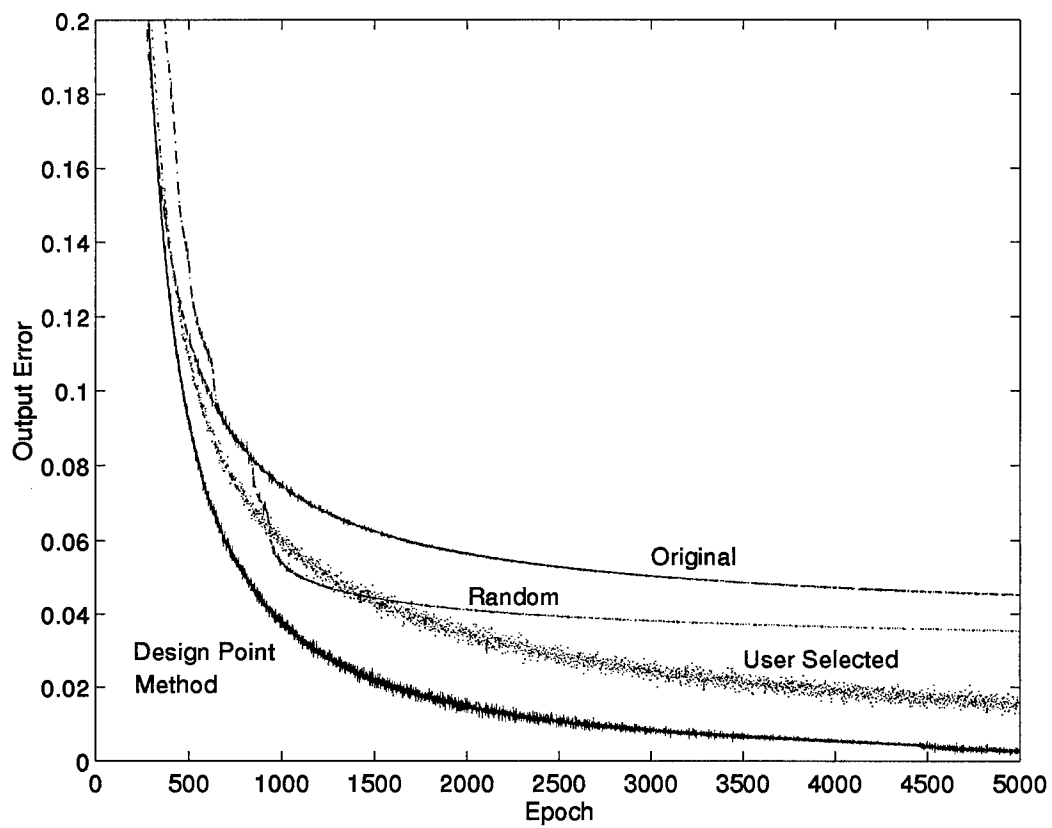
Figure 8. Linearly Separable Problem—Average Output Error Comparisons (30 Runs)
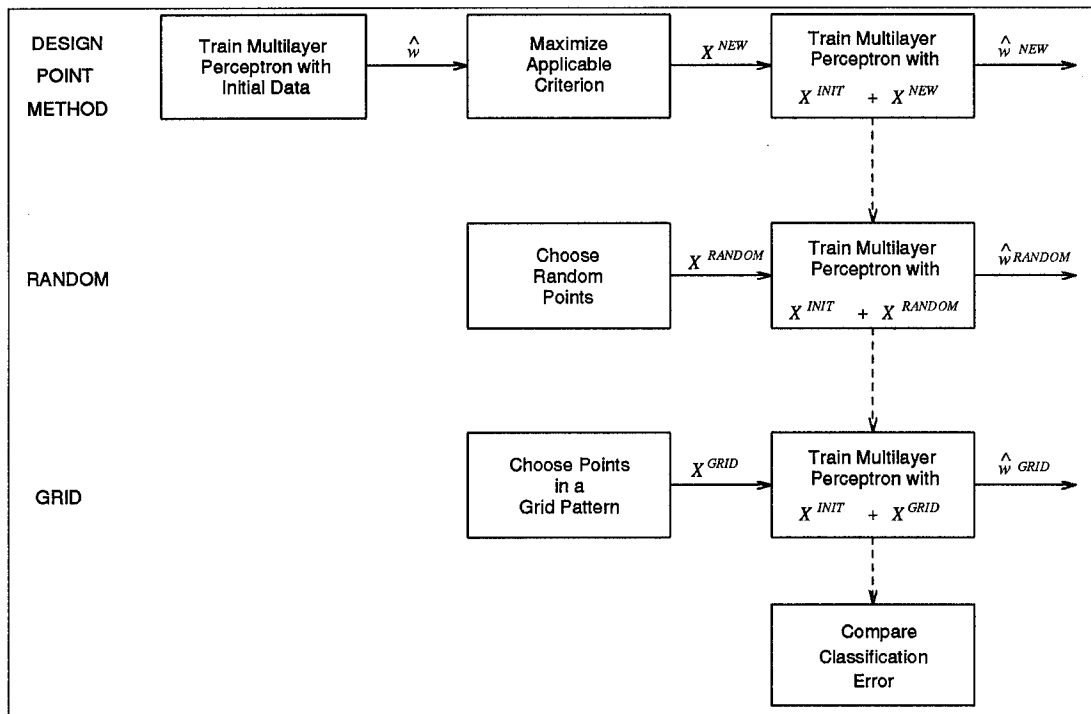
Figure 9. Research Method

set (X^{INIT}) to obtain preliminary weights ($\hat{w}$). Using these weights, the design point method

is used to determine new points where data should be gathered (X^{NEW}). Then, the multilayer

perceptron is trained with both data sets ($X^{INIT} + X^{NEW}$) to produce the final weight vector

($\hat{w}^{NEW}$).

Using randomly selected points requires no initial data set. However, for fair com-

parisons, a multilayer perceptron is trained with the initial data set and the random data set

($X^{INIT} + X^{RANDOM}$) to produce the final weight vector ($\hat{w}^{RANDOM}$). Similarly, when using

points chosen in a grid, the initial data set and the grided data set are combined to obtain

$\hat{w}^{GRID}$. Across these three methods, the average error rate on a test set is compared to judge

the accuracy of the resulting multilayer perceptrons.
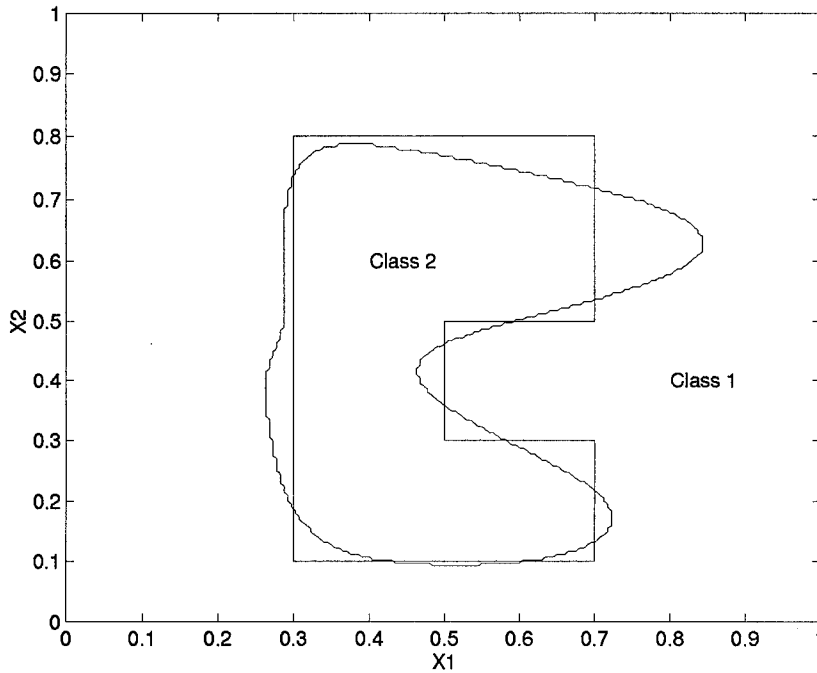
Figure 10. Nonlinearly Separable Problem—Truth Model and Original Multilayer Perceptron Boundary

3.3.3 Nonlinearly Separable Continuous Feature Space. The previous example is considered extremely easy for a multilayer perceptron. A more challenging discrimination problem is shown in Figure 10. In this example, the number of hidden nodes required to achieve an average classification accuracy of approximately 90 percent was eight. Inevitably, as the difficulty of the problem increases, the minimum number of hidden nodes increases *and* the dimensionality of the design point method increases. $|\hat{F}.|$ is a function of pn variables where p is the number of parameters (here, equal to the number of design points to be chosen) and n is the dimensionality of the input vector. Therefore, for this classification problem, $|\hat{F}.|$ is a function of $33 \cdot 2 = 66$ variables.

The multilayer perceptron was trained with 100 randomly selected training vectors to obtain the initial weight vector $\hat{w}$. Figure 10 also shows the original multilayer perceptron boundary defined by $\hat{w}$. The goal is to determine the best exemplars at which to run 33 future
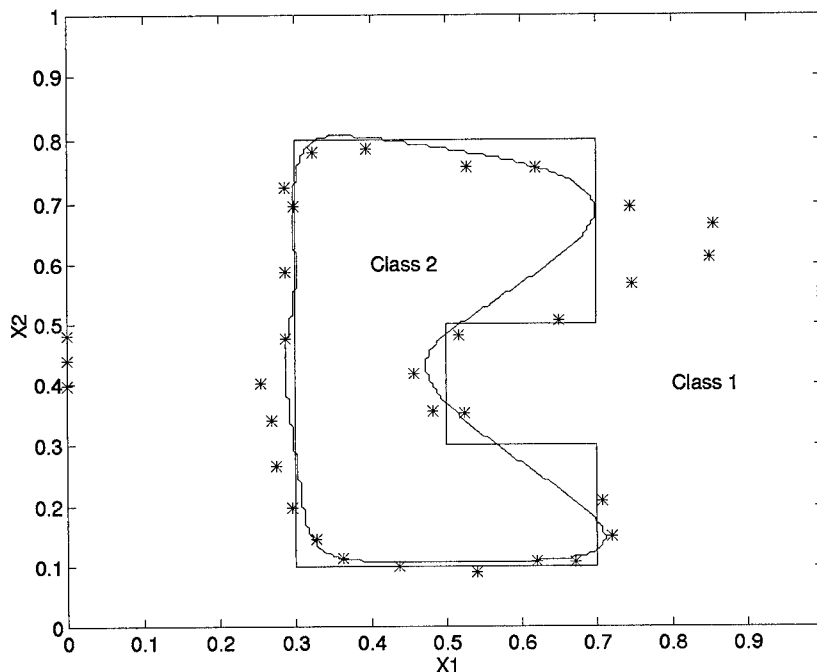
Figure 11. Nonlinearly Separable Problem—Design Points and Resulting Multilayer Perceptron Boundary

experiments. Powell's algorithm was used to maximize the determinant of the 33×33 matrix $\hat{F}$.. The resulting design points were classified according to the truth model and added to the training set. The multilayer perceptron was then retrained with the entire set of 133 exemplars. Figure 11 shows the design points and the new multilayer perceptron boundary.

For purposes of comparison, the multilayer perceptron was also trained with 33 randomly chosen points added to the training set. In addition, the multilayer perceptron was trained with a training set consisting of the initial 100 vectors and 36 points from a 6×6 grid across the feature space. The resulting test set classification error averaged over 30 runs of the multilayer perceptron for each case is shown in Figure 12.

Of course, the original training set of 100 vectors yielded the highest test set classification error (averaged over 30 runs) since one would expect that adding training vectors should only
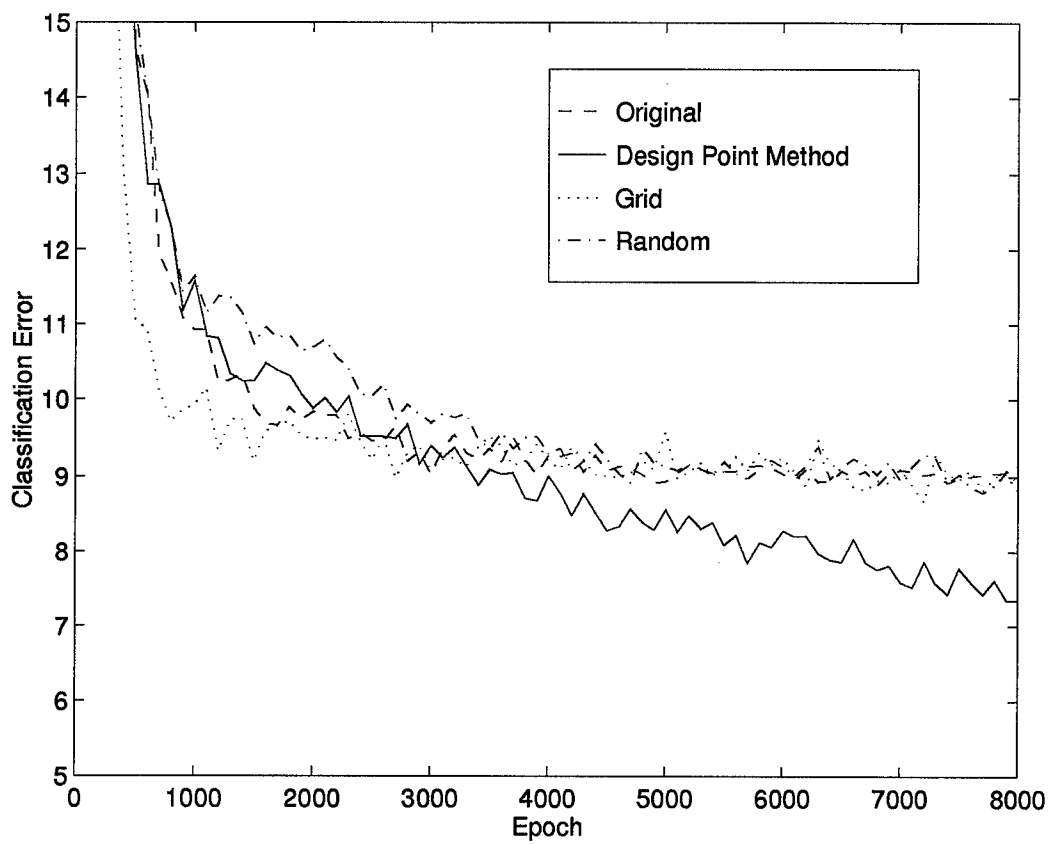
Figure 12. Nonlinearly Separable Problem—Average Classification Error Comparisons (30 Runs, Sampled Every 100 Epochs)

increase the accuracy of the classifier. The design point method test set error rate was the lowest of any of the data sets. The randomly chosen set and the grid set increased the accuracy of the multilayer perceptron, but not to the level of the design point method. The method described in Section 1.2.2 was used to determine that points chosen by the design point method yielded significantly lower error rates than points chosen randomly and points from the grid ($\alpha = 0.02$).

Run-time is a serious consideration when applying methods to multilayer perceptrons. Many applications require large numbers of inputs and large numbers of hidden nodes resulting in many, many weights. Below, the calculations required for implementing Powell's method to select design points are examined closer.

As stated earlier, Powell's method will converge in $\mathcal{N}$ loops if the function is quadratic and more than $\mathcal{N}$ loops if it is not. Here, $\mathcal{N}$ is the number of variables in the function to be maximized. $F.$ is a function of n (the dimension of the inputs) times N (the number of design points required) variables. Therefore, at least nN loops of Powell's algorithm are needed.

Within each loop of Powell's algorithm, the determinant $|F.^T F.|$ is calculated several times as the maximum is found in the direction of the current search. Calculation of a determinant using LU decomposition requires $\frac{1}{3}p^3$ executions (each execution consists of one multiply and one add) for a $p \times p$ matrix. The elements of $F.$ must be calculated each time Powell's algorithm requires evaluation of the determinant. Recall that the elements of $F.$ are partial derivatives $\frac{\partial z}{\partial w}$. The calculation of these derivatives requires calculation of the output of the multilayer perceptron for the particular exemplar under consideration. Depending on the size of the multilayer perceptron, this calculation could be expensive in terms of time.

Combining the calculation of multilayer perceptron outputs, determinants, and the overall calculations required for Powell's algorithm, one can see that the time required to

obtain design points could be extensive. The example problem discussed above required 51.5 minutes of system time (using a Sparc Sun station 5) to converge to a set of design points.

3.3.4   Ranking.   The set of 33 vectors selected for the problem above may be too large. The user has two choices:

1. Re-accomplish the entire design point method selecting some lesser number of design points.

2. Rank order the 33 chosen points as to their usefulness in determining the optimal parameter set. Two possible ranking procedures are outlined below.

The first of these choices may require a significant amount of computation depending on the size of the multilayer perceptron. The benefit of the second choice is that the maximization (which may be numerically complex and time consuming) is performed once. Then, the experimenter is free to choose the number of design points needed and add other design points as resources permit. It may also be that the design points are required immediately with no time to perform the optimization again. In addition, if resources are scarce, ordering the design points assures that the "most important" experiments are performed first. The disadvantage of ranking is that a subset of the selected design points is a sub-optimal set.

Method 1—Dot Product Ranking. The discrete design point algorithm covered in Section 3.2.2 suggests a method for ranking design points [44]. Recall that the algorithm chooses a design point for inclusion from a set of feasible points if it maximizes $\hat{\mathbf{f}}.^{T}(\hat{F}.^{T}\hat{F}.)^{-1}\hat{\mathbf{f}}.$. *Existing* design points can be ranked similarly.

A stated earlier, choosing the exemplar that maximizes $\hat{\mathbf{f}}.^{T}(\hat{F}.^{T}\hat{F}.)^{-1}\hat{\mathbf{f}}.$ ensures a maximum value of $|F.^{T}F.|$. However, the quantity $\hat{\mathbf{f}}.^{T}(\hat{F}.^{T}\hat{F}.)^{-1}\hat{\mathbf{f}}.$ has an alternative inter-

pretation. Given the nonlinear model $y = f(\mathbf{x}; \boldsymbol{\theta}) + \varepsilon$, let

$$y_0 = f(\mathbf{x}_0; \boldsymbol{\theta}) + \varepsilon_0 \tag{86}$$

where $\mathbf{x}_0$ is any point in $\mathcal{R}$, y_0 is the true response at x_0 and $\varepsilon_0 \sim N(0, \sigma^2)$ and is independent of ε. An estimate of y_0 is given by the fitted model as $\hat{y}_0 = f(\mathbf{x}_0; \hat{\boldsymbol{\theta}})$. The Taylor expansion is

$$f(\mathbf{x}_0; \hat{\boldsymbol{\theta}}) \approx f(\mathbf{x}_0; \boldsymbol{\theta}) + \mathbf{f}_{\cdot 0}^T (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \tag{87}$$

where

$$\mathbf{f}_{\cdot 0}^T = \left(\frac{\partial f(\mathbf{x}_0; \boldsymbol{\theta})}{\partial \theta_1}, \frac{\partial f(\mathbf{x}_0; \boldsymbol{\theta})}{\partial \theta_2}, \cdots, \frac{\partial f(\mathbf{x}_0; \boldsymbol{\theta})}{\partial \theta_p} \right) \tag{88}$$

Then,

$$\begin{aligned} y_0 - \hat{y}_0 &\approx y_0 - f(\mathbf{x}_0; \boldsymbol{\theta}) - \mathbf{f}_{\cdot 0}^T (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \\ &= \varepsilon_0 - \mathbf{f}_{\cdot 0}^T (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \end{aligned} \tag{89}$$

and since $\hat{\boldsymbol{\theta}} \sim N_p(0, \sigma^2 (F_{\cdot}^T F_{\cdot})^{-1})$,

$$\begin{aligned} \text{var}[y_0 - \hat{y}_0] &\approx \text{var}[\varepsilon_0] + \text{var}[\mathbf{f}_{\cdot 0}^T (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})] \\ &\approx \sigma^2 + \sigma^2 \mathbf{f}_{\cdot 0}^T (F_{\cdot}^T F_{\cdot})^{-1} \mathbf{f}_{\cdot 0} \\ &= \sigma^2 (1 + \mathbf{f}_{\cdot 0}^T (F_{\cdot}^T F_{\cdot})^{-1} \mathbf{f}_{\cdot 0}) \end{aligned} \tag{90}$$

This variance of the estimated response can be calculated using $\boldsymbol{\theta}$:

$$\text{var}[y_0 - \hat{y}_0] \approx \sigma^2 (1 + \hat{\mathbf{f}}_{\cdot 0}^T (\hat{F}_{\cdot}^T \hat{F}_{\cdot})^{-1} \hat{\mathbf{f}}_{\cdot 0}) \tag{91}$$

So, choosing the exemplar $\mathbf{x}_0$ that maximizes $\hat{\mathbf{f}}_{\cdot 0}^T (\hat{F}_{\cdot}^T \hat{F}_{\cdot})^{-1} \hat{\mathbf{f}}_{\cdot 0}$ is equivalent to choosing the point at which the variance of the predicted response is a maximum [59].

One might consider ranking each exemplar, x_0, with $\hat{\mathbf{f}}_{.0}^T(\hat{F}.^T\hat{F}.)^{-1}\hat{\mathbf{f}}_{.0}$ using the entire set of design points to calculate $(\hat{F}.^T\hat{F}.)$. However, as shown below, this approach cannot be used. Without loss of generality, assume that $\hat{\mathbf{f}}_{.0}^T(\hat{F}.^T\hat{F}.)^{-1}\hat{\mathbf{f}}_{.0}$ is evaluated at the exemplar corresponding to the first row of $\hat{F}.$. The following results if $F.$ is square and $F.^{-1}$ exists.

$$
\begin{aligned}
\hat{\mathbf{f}}_{.0}^T(\hat{F}.^T\hat{F}.)^{-1}\hat{\mathbf{f}}_{.0} &= \hat{\mathbf{f}}_{.0}^T\hat{F}.^{-1}(\hat{F}.^T)^{-1}\hat{\mathbf{f}}_{.0} \\
&= \left[\hat{F}.\hat{F}.^{-1}\right]_{1j}\left[(\hat{F}.^T)^{-1}\hat{F}.^T\right]_{j1} \quad j = 1,\ldots,p \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\
&= 1 \hspace{4cm} (92)
\end{aligned}
$$

Note that $\mathrm{var}[y_0 - \hat{y}_0]$ in this case is $2\sigma^2$. Exactly the same results are obtained for the other exemplars used to calculate $(F.^TF.)^{-1}$.

Clearly then, one cannot in all circumstances use a variance-covariance matrix estimated from the design points to rank that same set of design points. In this research, the variance-covariance matrix will be assumed equal to the identity and the following ranking measure established:

$$
\mathcal{M}_1(\mathbf{x}) = \hat{\mathbf{f}}.^T(\mathbf{x})\hat{\mathbf{f}}.(\mathbf{x}) \hspace{3cm} (93)
$$

where $\mathbf{x} = (x_1, x_2, \ldots x_n)$ is the design point under consideration. In multilayer perceptron notation,

$$
\hat{\mathbf{f}}.^T(\mathbf{x}) = \left(\frac{\partial z(\mathbf{x}; \mathbf{w})}{\partial w_1}, \frac{\partial z(\mathbf{x}; \mathbf{w})}{\partial w_2}, \ldots, \frac{\partial z(\mathbf{x}; \mathbf{w})}{\partial w_p}\right) \hspace{2cm} (94)
$$

The exemplar in the design set having the greatest value of $\mathcal{M}_1$ will be ranked as best. The measure $\mathcal{M}_1$ can also be interpreted as the squared length of the gradient vector evaluated at **x**.

Method 2—Saliency Ranking. A second method of ranking design points is suggested by the saliency measure introduced in Section 1.3. Once D-optimal design points have been selected, it makes sense to choose the points in the set which effect the greatest change in the output of the multilayer perceptron. The saliency measure was originally presented as a method of ranking features rather than ranking exemplars consisting of several features. Therefore, the original measure will be modified. The saliency measure is based on $\frac{\partial z_j}{\partial x_k}$. To find exemplars having the greatest effect on z_j, the total change in z_j due to changes in x_k should be maximized. This total change in z_j may be represented by the total differential

$$dz_j = \frac{\partial z_j}{\partial x_1}dx_1 + \cdots + \frac{\partial z_j}{\partial x_i}dx_i + \cdots + \frac{\partial z_j}{\partial x_n}dx_n \tag{95}$$

and, by expanding the partial derivatives for the multilayer perceptron

$$dz_j = \sum_k z_j(1 - z_j) \sum_i w_{ij}^2 x_i^1(1 - x_i^1)w_{ki}^1 dx_k \tag{96}$$

In this setting, total differentials are being compared between exemplars so that dx_k will be set to some constant for all k.

So far, only a single multilayer perceptron output has been considered. Including all outputs the following measure is defined

$$\mathcal{M}_2(\mathbf{x}) = \sum_{j=1}^{r} \sum_{k=1}^{n} \left| z_j(1 - z_j) \sum_{i=1}^{m} w_{ij}^2 x_i^1(1 - x_i^1)w_{ki}^1 \right| \tag{97}$$

59

Table 3. Comparing Ranking Measures—Final Average Test Set Classification Errors

Method	Set Added to Training Set	Average Test Set Classification Error
Dot Product ($\mathcal{M}_1$)	Ten Best Exemplars	8.68
	Ten Worst Exemplars	9.34
Saliency ($\mathcal{M}_2$)	Ten Best Exemplars	8.78
	Ten Worst Exemplars	10.55

where x is the design point under consideration and current weight estimates $\hat{w}$ are used. Design points will be ordered according to this measure with large values corresponding to "good" design points.

Initially, the 33 design points for the nonlinearly separable problem above were rank ordered according to Method 1. In order to judge the usefulness of the design points, ten vectors with the largest values of the measure were grouped as a design point set and ten vectors with the smallest values of the measure were grouped in a design point set. Two multilayer perceptrons were trained—one with the vectors with large values added to the training set and one with the vectors with small values added to the training set. Figure 13 shows the design points and the value of $\mathcal{M}_1$ for each design point. The final classification error averaged over 30 runs is shown in Table 3. The difference in the average classification errors for the ten best and ten worst points is statistically significant.

Next, Method 2 was used to separate the 33 design points into a "high" and "low" set of ten vectors each. Figure 14 shows the design points and the value of $\mathcal{M}_2$ for each design point. Figure 3 shows the average classification error for the new multilayer perceptrons. Again, the difference in the average classification errors is statistically significant.

60
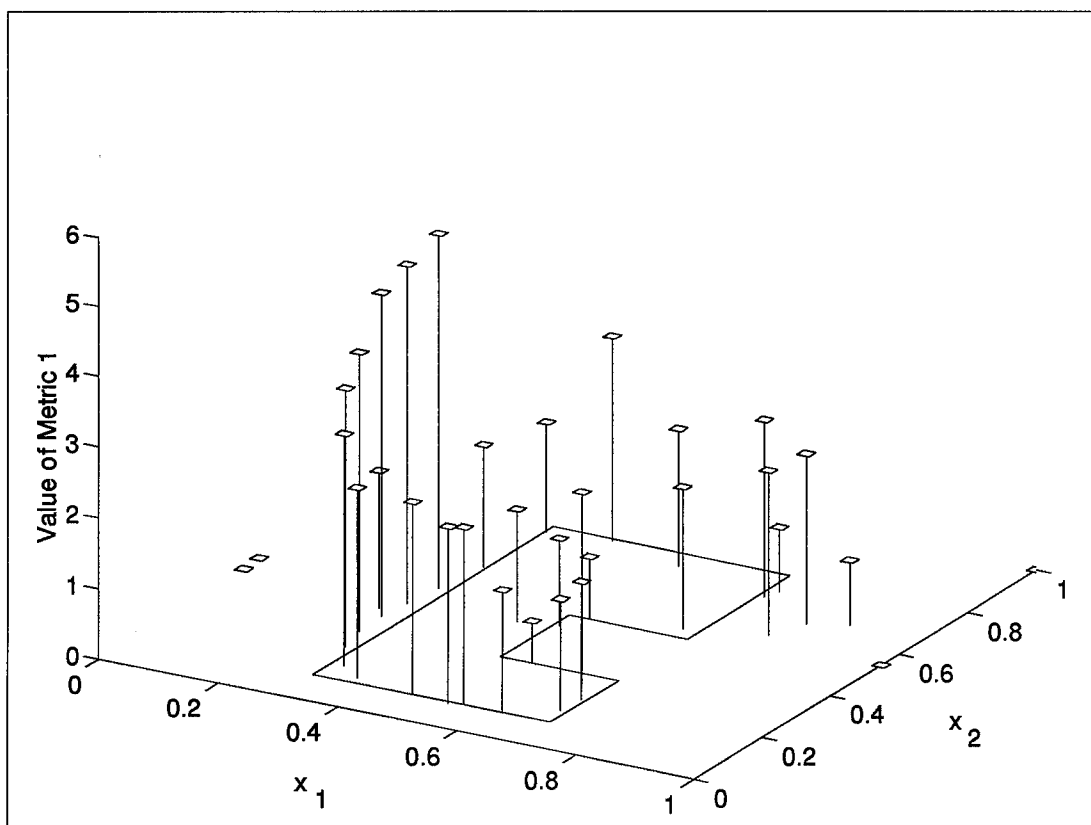
Figure 13. Design Points and $\mathcal{M}_1$
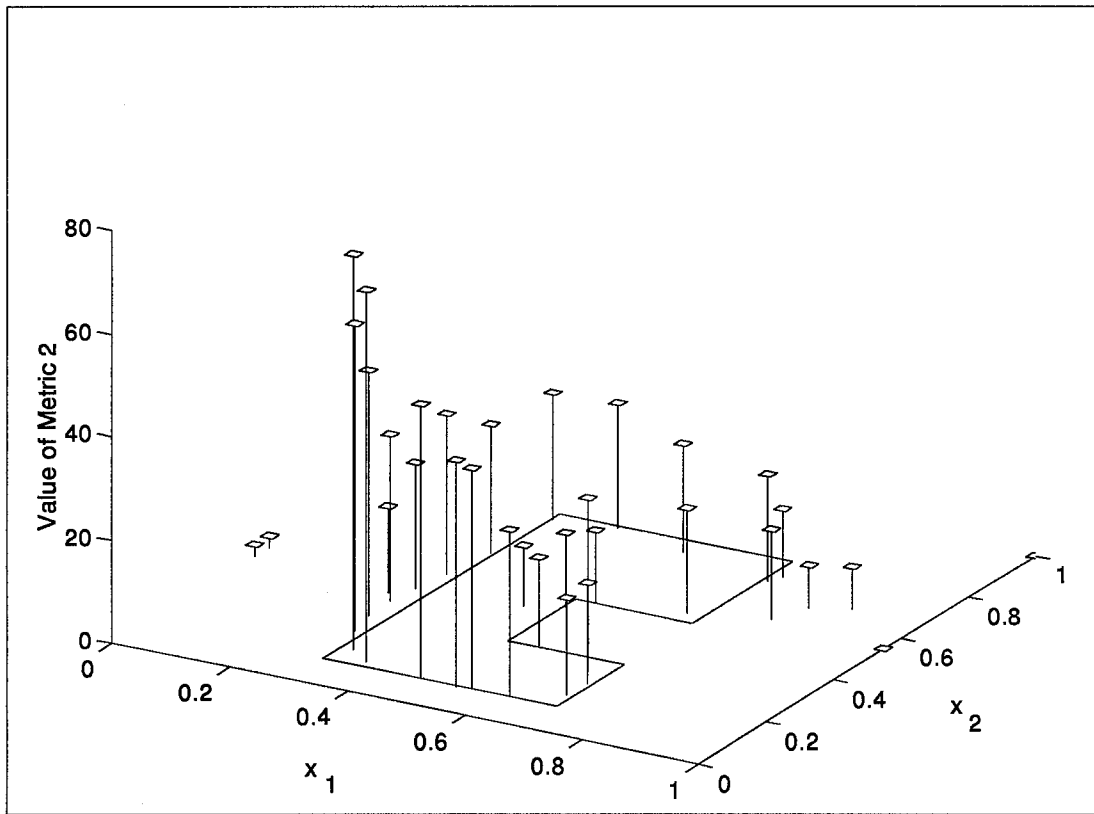
Figure 14. Design Points and $\mathcal{M}_2$

3.3.5 Nonlinearly Separable Discrete Feature Space. In this section, the differences and similarities of using a discrete as opposed to continuous feature space are demonstrated. The same discrimination problem used in the last section will be used here.

Implementing the discrete exchange algorithm, 33 design points were chosen from a feasible set of 400 points. Approximately 30 exchanges were required in the discrete exchange algorithm to arrive at an optimal set. The design points are shown in Figure 15. A multilayer perceptron was trained using the initial training set and the 33 additional points. The resulting average test set classification error rate is compared to the classification error rate given a continuous feature space (see Section 3.3.3, Figure 12) in Figure 16. The discrete method is also compared to the error rate obtained when adding 33 randomly chosen points to the training set. Using the 30 runs of the multilayer perceptron in each case, the mean error rate was significantly lower for the discrete design method as compared to the random point set ($\alpha = 0.05$). Observe that the discrete method did not improve the classification error as much as the continuous method. One might expect this since the discrete method allows for the selection of only a subset of the points available to the continuous method.

In comparing the discrete and continuous methods, the time expended in obtaining the design points should also be considered. The continuous method requires considerably more computer processing time than the discrete method. For the problem considered here, the continuous method required approximately 51.5 minutes of system time (on a Sun Sparc station 5) while the discrete method required only 27.1 seconds of system time for one run.

3.4 Reducing Complexity of Design Point Determination

If multilayer perceptrons with many inputs or many hidden nodes are required for a particular classification task, determining the design points may be difficult. As stated previously, the number of loops required in Powell's algorithm is a function of the dimension
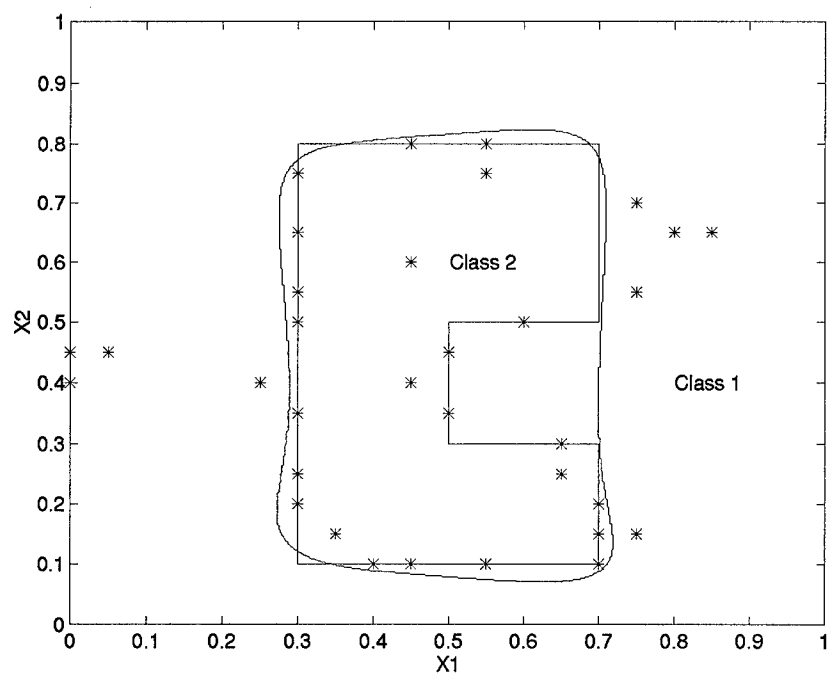
Figure 15. Nonlinearly Separable Problem—Design Points and Resulting Boundary (Discrete Feature Space)
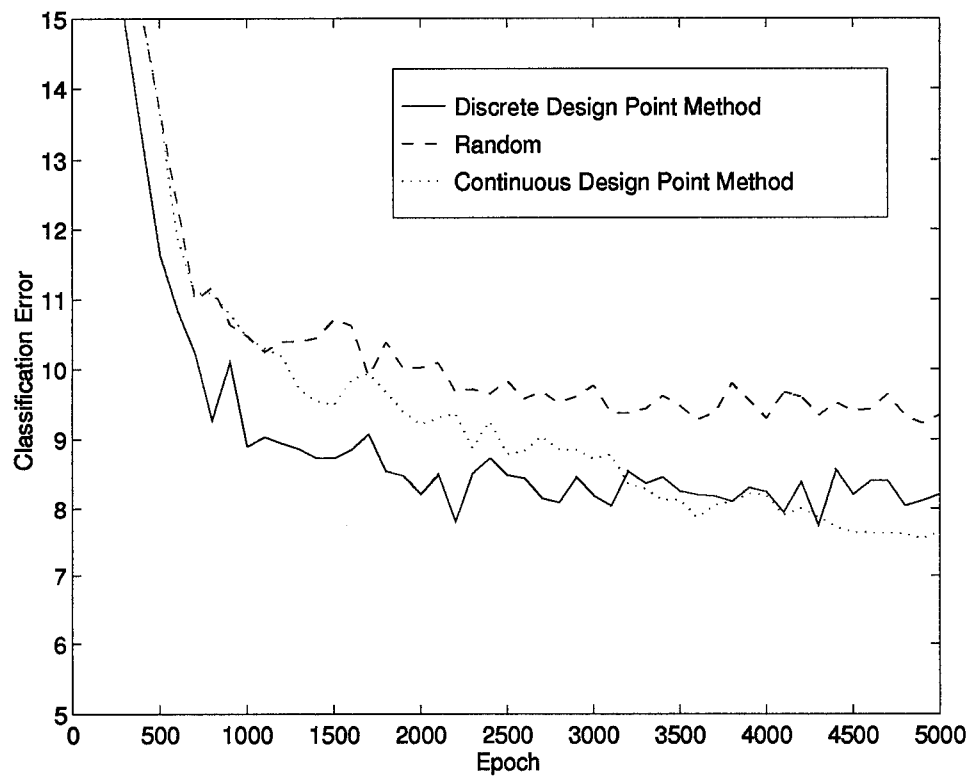
Figure 16. Nonlinearly Separable Problem—Average Classification Error Comparison for Discrete Feature Space (30 Runs, Sampled Every 100 Epochs)

of the input vector and the desired number of design points. Also, the determinant of the F. matrix is a function of the number of weights times the number of inputs. This potentially large number of variables presents two problems for the maximization routines:

1. Large Number of Parameters (Multilayer Perceptron Weights)

2. Smallness of Partial Derivatives in the F. Matrix

Examples from the nonlinear regression literature illustrate results for much smaller problems than were attempted here. For example, Johnson and Nachtsheim compare algorithms for constructing D-optimal designs for nonlinear regression problems and only test up to $n=15$ [35].

The nonlinearly separable discrimination problem addressed in the previous section is a reasonably large problem. However, the possibility of larger network structures must be considered. Another example problem (not shown here) with four inputs and ten hidden nodes required 35.93 hours of system time to find 33 design points. The value determinant for this example was on the order of 10^{-700}. Clearly, some simplifications are required.

Originally, it was thought that one of the other optimality criterion could be used to reduce the complexity of finding design points. For example, calculating the trace of a matrix is computationally less complex than calculating the determinant. However, in the specific case being observed here, the trace is just as computationally complex. The reason stems from the simplifications that are possible when using the determinant. Similar reductions in complexity are not possible with the other criterion. It appears that alternate forms of simplification are required.

In this section, methods of simplifying the stated procedure to avoid computational difficulties are discussed. First, a method to reduce the size of the required multilayer

perceptron (and consequently the number of weights) is presented. Second, a simplified measure on only a subset of the parameters is presented.

3.4.1 Partially Nonlinear Models—Direct Linear Feedthrough (DLF) Networks. This section details a procedure to exploit linearities in a given discrimination problem. If linearities are present, a linear representation within the multilayer perceptron makes it possible to reduce the overall number of weights. This linear addition results in a Direct Linear Feedthrough (DLF) network. If a DLF network can be used, then it can be shown that the weights corresponding to the linear part of the network have no effect on the D-optimality of a certain design. The result is a reduction in the dimensionality of $|F.|$, a simplification of the elements of $F.$, and therefore, quicker convergence to design points that yield a maximum.

3.4.1.1 Direct Linear Feedthrough (DLF) Networks. Lee and Holt suggest the use of direct linear feedthroughs together with the more traditional multilayer feedforward network [39, 27]. The direct linear feedthrough (DLF) network is shown in Figure 17.

The DLF network is constructed by superposing the linear connections between input and output nodes to the standard multilayer network. If the sigmoid is no longer applied to the upper layer, then the output from this network structure is simply the summation of the output from the standard network and the linear network which is

$$z_j^{\text{total}} = z_j^{\text{net}} + z_j^{\text{linear}} \tag{98}$$

where j indicates the jth output node and

$$z_j^{\text{net}} = \sum_{i=1}^{m} w_{ij}^2 x_i^1 + \xi_j^2 \qquad j = 1, \ldots, r \tag{99}$$

67
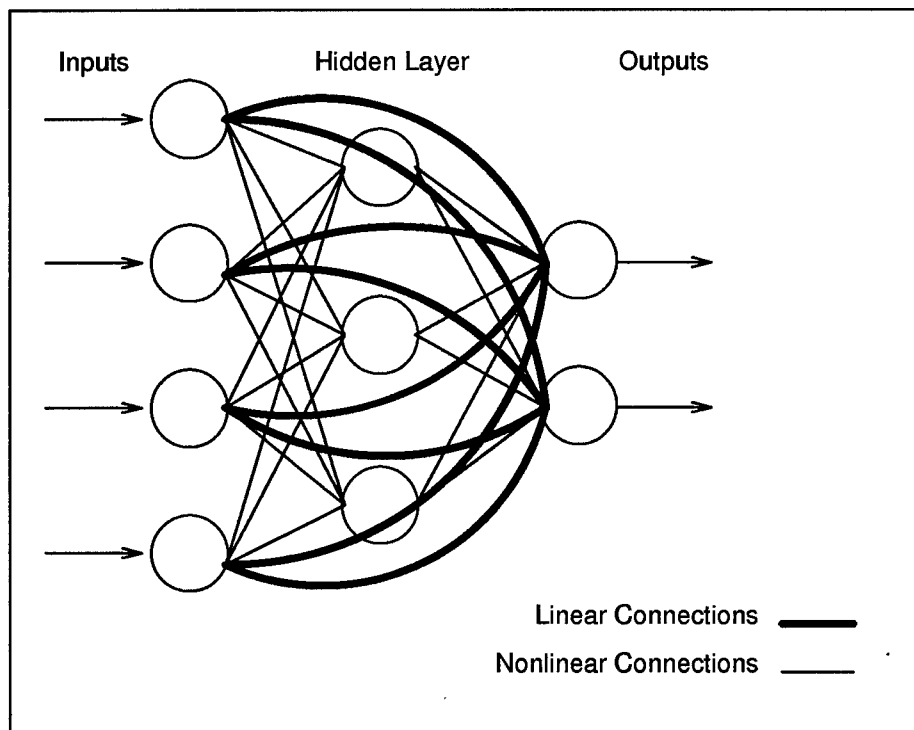
Figure 17. Direct Linear Feedthrough Network

Reprinted from [39]

and

$$z_j^{\text{linear}} = \sum_{k=1}^{n} w_{kj}^l x_k + \xi^l \qquad j = 1, \ldots, r \qquad (100)$$

The l superscript indexes the linear weights.

DLF network structure is most useful when the linear/nonlinear relationship between the input and output is unknown. When the functionality between the input and output is linear, the contribution from the nonlinear part of the DLF network will decrease until it is invisible to the network. When the relationship between the input and output is nonlinear, the linear and nonlinear pieces will work together to best model the data.

The weights of the linear part of the network may be obtained from linear regression techniques and preset within the network. Lee and Holt state that this initialization method can shorten the training and ensure the stability of the training [39]. Even if the linear weights are initialized with random numbers, they will quickly adjust themselves to an approximate least squares solution. This phenomena is due to the fact that the gradients of the objective function with respect to the linear weights are much steeper than nonlinear weights. Thus, the linear weights are adjusted at a faster rate at the beginning of the learning process [39]. For this reason, it may be necessary to use different learning rates for the linear and nonlinear parts of the network. In addition, now that a sigmoid is no longer used on the output, it may also be necessary to use different learning rates for the upper and lower weights. Training the DLF network requires a slightly modified version of backpropagation. The DLF network version is derived in Appendix C.

3.4.1.2 Designs for Partially Nonlinear Models. The advantage in using a DLF network becomes apparent when the number of hidden nodes required can be reduced by including linear connections. When there are linear terms in any nonlinear model, the

69

design of experiments criterion is simplified. Hill discusses this simplification by defining a "partially nonlinear" model and stating a theorem concerning partially nonlinear models [29].

Definition 1 *We say that a regression model* $\eta(\mathbf{x}, \boldsymbol{\theta})$ *is* **partially nonlinear** *if the* $p \times 1$ *derivative vector* $\mathbf{f}.(\mathbf{x}, \boldsymbol{\theta}) = \frac{\partial \eta(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ *can be represented as*

$$\mathbf{f}.(\mathbf{x}; \boldsymbol{\theta}) = A(\boldsymbol{\theta})\mathbf{g}(\mathbf{x}, \boldsymbol{\phi}) \tag{101}$$

where $A(\boldsymbol{\theta})$ *is a nonsingular matrix not involving* $\mathbf{x}$*, and* $\mathbf{g}(\mathbf{x}, \boldsymbol{\phi})$ *is a vector of functions depending on* $\mathbf{x}$ *and on a subset* $\boldsymbol{\phi}$ *of certain of the components* $\boldsymbol{\theta}$*. The components of* $\boldsymbol{\theta}$ *represented by* $\boldsymbol{\phi}$ *are those which appear nonlinearly in* $\eta(\mathbf{x}, \boldsymbol{\theta})$ *[29].*

Theorem 3 *Consider a partially nonlinear regression model* $\eta(\mathbf{x}, \boldsymbol{\theta})$*, for which*

$$\mathbf{f}.(\mathbf{x}, \boldsymbol{\theta}) = A(\boldsymbol{\theta})\mathbf{g}(\mathbf{x}, \boldsymbol{\phi}) \tag{102}$$

as in the definition above. The D-optimal design for the parameter $\boldsymbol{\theta}$ *in this model depends on those components of* $\boldsymbol{\theta}$ *which are in the sub-vector* $\boldsymbol{\phi}$ *but not on the remaining components.*

Proof:

$$
\begin{aligned}
F.^{T}(\boldsymbol{\theta})F.(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \mathbf{f}.(\mathbf{x}_i; \boldsymbol{\theta})\mathbf{f}.^{T}(\mathbf{x}_i; \boldsymbol{\theta}) \tag{103} \\
&= \sum_{i=1}^{n} A(\boldsymbol{\theta})\mathbf{g}(\mathbf{x}_i; \boldsymbol{\phi})\mathbf{g}^{T}(\mathbf{x}_i; \boldsymbol{\phi})A^{T}(\boldsymbol{\theta}) \\
&= A(\boldsymbol{\theta}) \left[\sum_{i=1}^{n} \mathbf{g}(\mathbf{x}_i; \boldsymbol{\phi})\mathbf{g}^{T}(\mathbf{x}_i; \boldsymbol{\phi}) \right] A^{T}(\boldsymbol{\theta}) \tag{104}
\end{aligned}
$$

Therefore,

$$|F.^{T}(\boldsymbol{\theta})F.(\boldsymbol{\theta})| = |A(\boldsymbol{\theta})|^{2} |\sum_{i=1}^{N} \mathbf{g}(\mathbf{x}_i; \boldsymbol{\phi})\mathbf{g}^{T}(\mathbf{x}_i; \boldsymbol{\phi})| \tag{105}$$

so that the choice of $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ *to maximize* $|F_{.}^{T}(\boldsymbol{\theta})F_{.}(\boldsymbol{\theta})|$ *depends on only the* ϕ *components of* $\boldsymbol{\theta}$ *since* $\mathbf{g}(\mathbf{x}_i; \phi)$ *does not involve the other components* [29].

For the DLF network, $f_{.}(\mathbf{x}; \boldsymbol{\theta}) = I\mathbf{g}(\mathbf{x}; \phi)$ with the D-optimal design for $\boldsymbol{\theta}$ independent of the weights connecting the inputs to the outputs.

So, when using a DLF network, one can use the simplification above and not include the parameters associated with the linear part of the model. Two benefits are realized. First, if there are linearities in the classification problem, the DLF network allows one to develop a classifier with fewer hidden nodes and fewer weights overall. Secondly, the elements of $F_.$ associated with the linear parts of the DLF network are very simple. If p_n is the number of nonlinear weights and p_l is the number of linear weights, then the structure of $F_.$ matrix for a DLF network is:

$$
F_{.}^{\mathrm{DLF}} = \left[\begin{array}{cccc|cccc}
\frac{\partial z^1}{\partial w_1} & \frac{\partial z^1}{\partial w_2} & \cdots & \frac{\partial z^1}{\partial w_{p_n}} & x_{1,p_n+1} & x_{1,p_n+2} & \cdots & x_{1,p_n+p_l} \\[2mm]
\frac{\partial z^2}{\partial w_1} & \frac{\partial z^2}{\partial w_2} & \cdots & \frac{\partial z^2}{\partial w_{p_n}} & x_{2,p_n+1} & x_{2,p_n+2} & \cdots & x_{2,p_n+p_l} \\[1mm]
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\[1mm]
\frac{\partial z^{p_n}}{\partial w_1} & \frac{\partial z^{p_n}}{\partial w_2} & \cdots & \frac{\partial z^{p_n}}{\partial w_{p_n}} & x_{p_n,p_n+1} & x_{p_n,p_n+2} & \cdots & x_{p_n,p_n+p_l} \\[1mm]
\hline
\frac{\partial z^{p_n+1}}{\partial w_1} & \frac{\partial z^{p_n+1}}{\partial w_2} & \cdots & \frac{\partial z^{p_n+1}}{\partial w_{p_n}} & x_{p_n+1,p_n+1} & x_{p_n+1,p_n+2} & \cdots & x_{p_n+1,p_n+p_l} \\[1mm]
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\[1mm]
\frac{\partial z^{p_n+p_l}}{\partial w_1} & \frac{\partial z^{p_n+p_l}}{\partial w_2} & \cdots & \frac{\partial z^{p_n+p_l}}{\partial w_{p_n}} & x_{p_n+p_l,p_n+1} & x_{p_n+p_l,p_n+2} & \cdots & x_{p_n+p_l,p_n+p_l}
\end{array}\right]
$$

$$(106)$$

This matrix is $(p_n + p_l) \times (p_n + p_l)$. Note that none of the linear weights are required and that two of the blocks in the matrix are simply matrices of the design points. Because the sigmoid is no longer applied to the output, the derivatives in the $F_.$ matrix are now

$$
\frac{\partial z^s}{\partial w_{ki}^1} = w_{ij}^2 x_i^{1s}(1 - x_i^{1s})x_k^s \tag{107}
$$

71

and

$$\frac{\partial z^s}{\partial w_{i1}^2} = x_i^{1s} \qquad (108)$$

(See Equations 77 and 78 for the original forms of these derivatives.)

3.4.1.3   Results—Sample Problem. Given any classification problem, some framework must be used for testing whether the DLF structure will yield a simplification. Figure 18 shows one possible method for determining whether a DLF network would be used and, if so, how many hidden nonlinear nodes should be used.

This framework was used on the classification problem shown in the nonlinear problem of Section 3.3.3, Figure 10 to determine that the optimal structure is a DLF network with four hidden nodes. This network contains 20 weights. In this example, the acceptable classification error rate was purposely chosen high (average $\mathcal{E}_C$ of less than 15 percent over 30 runs). This was done to show that it is possible to use weights from an imperfectly trained multilayer perceptron and still obtain reasonable design points.

Using the weights from the DLF network, design points were chosen with the matrix to be maximized in the form of Equation 106. The design points and the initial DLF network boundary are shown in Figure 19.

These design points were added to the initial training set and the DLF network was retrained 30 times. The best resulting DLF network activation and boundary is shown in Figure 20. The average test set classification error over 30 runs improved from 13.68 percent (before the design points were added to the training set) to 12.14 percent (after the design points were added) as shown in Figure 21. This is compared with an average test set classification error of 14.30 for randomly selected points.

In summary, the DLF network can be a useful tool whenever simplification of a multilayer perceptron is required. The overall number of weights can be reduced and the activations

72
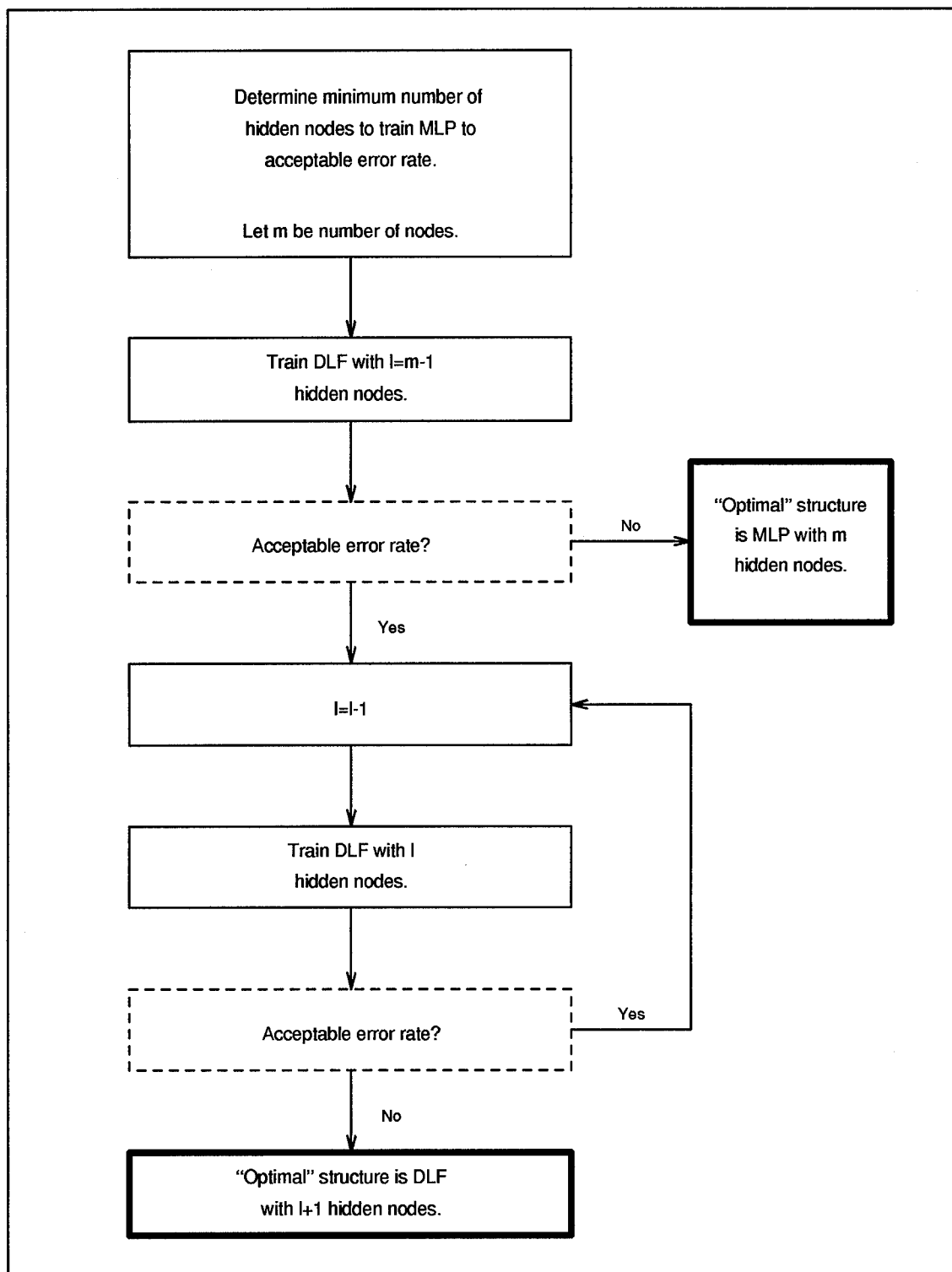
Figure 18. Determining DLF Structure
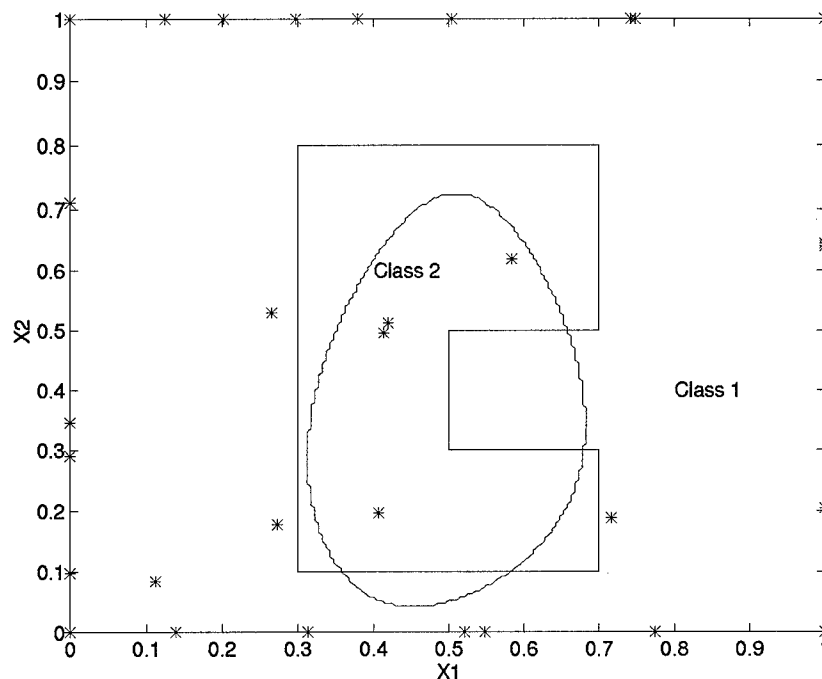
Figure 19. Design Points and Initial Decision Boundary for DLF Network

simplified. The use of the DLF network is especially significant in the design of experiments arena, since the complexity of the design point method can be greatly reduced.

3.4.2 Subsets of Parameters—Using Lower Layer Weights. A second method of reducing the complexity of determining design points may be to consider only a subset of the parameters. In a multilayer perceptron it may be possible to choose design points to best estimate the lower layer weights while not considering the upper layer weights.

3.4.2.1 Optimal Designs for Subsets of Parameters. Hill and Hunter present a method of planning experiments to estimate a subset of the parameters in the model. Let θ be

74
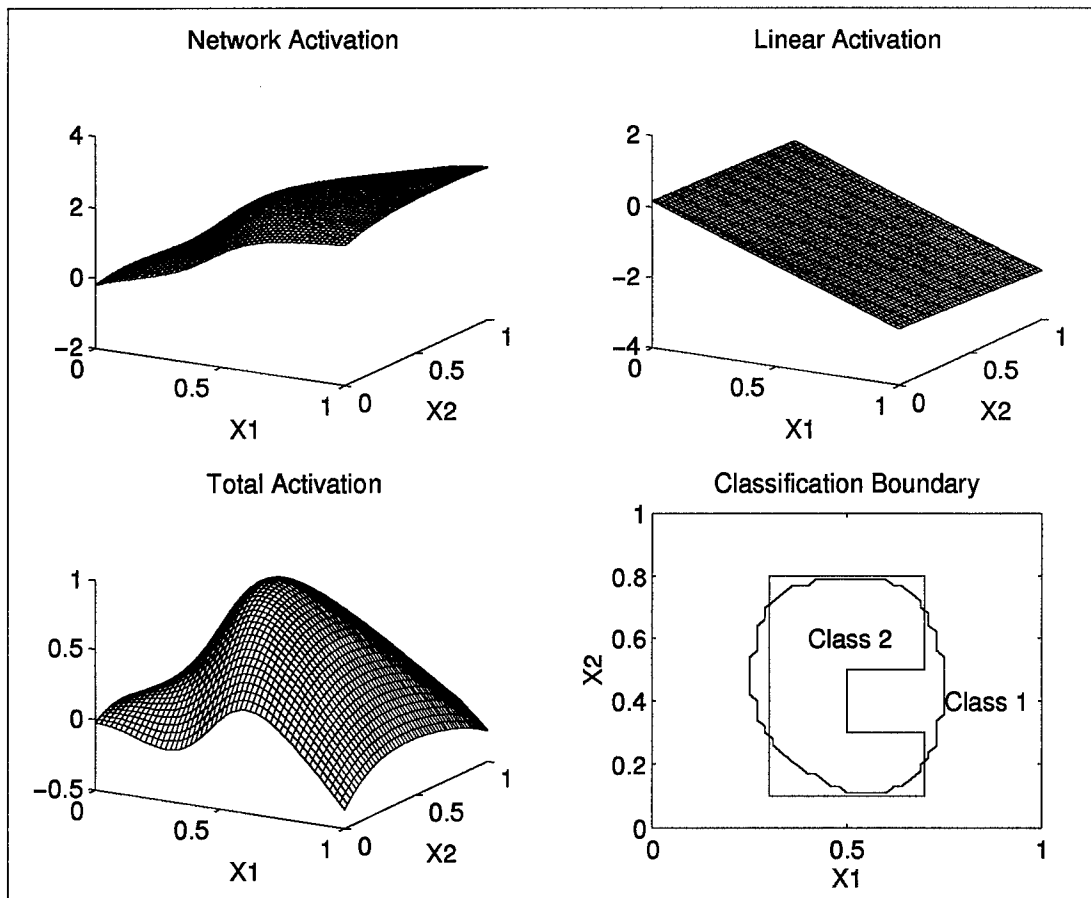
Figure 20. Resulting Activations and Decision Boundary for DLF Network
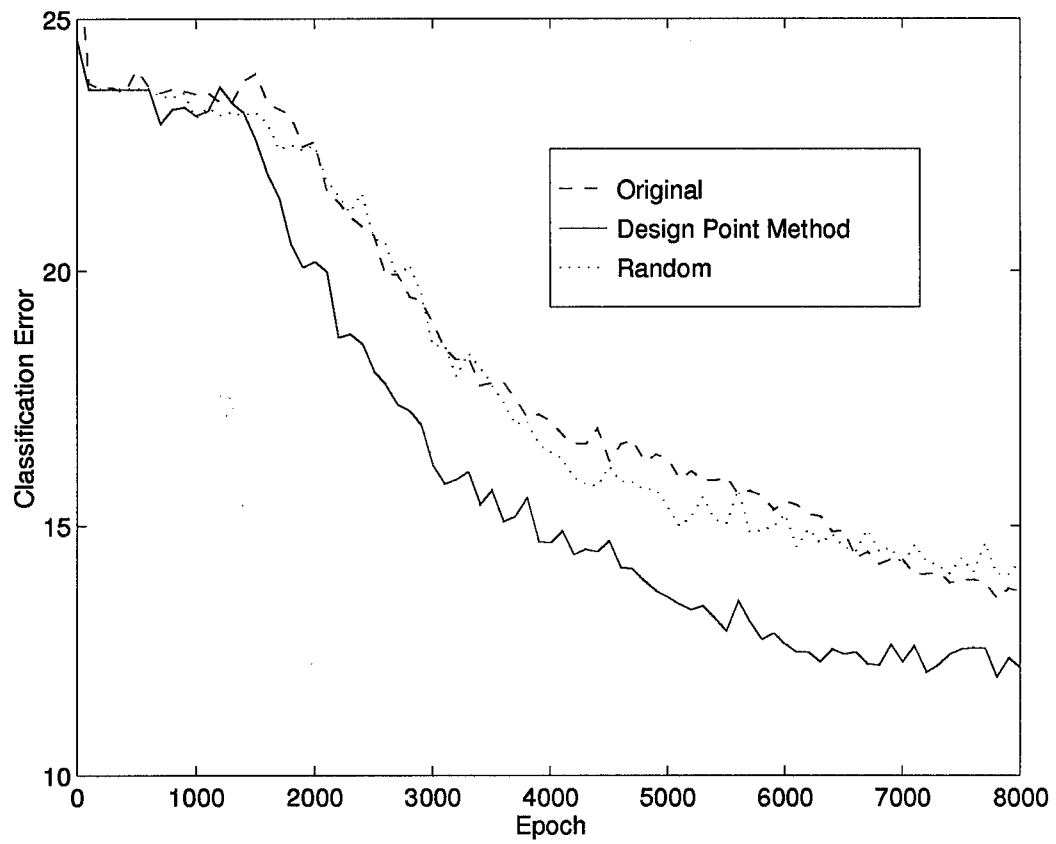
Figure 21. Average Classification Error Comparison for DLF Network (30 Runs, Sampled Every 100 Epochs)

partitioned as

$$\theta = \begin{bmatrix} \theta_1 \\ \dots \\ \theta_2 \end{bmatrix} \tag{109}$$

where θ is $p \times 1$, θ_1 is $q \times 1$ and contains the parameters of interest, θ_2 is $u \times 1$ and contains those parameters of less interest. Then, let

$$F. = \begin{bmatrix} F_{\cdot 1} & \vdots & F_{\cdot 2} \end{bmatrix} \tag{110}$$

where $F_{\cdot 1}$ is $N \times q$ and $F_{\cdot 2}$ is $N \times u$.

Seber and Wild state that an approximate confidence region for θ_1 may be written as

$$\{\theta_1 : (\theta_1 - \hat{\theta}_1)^T F_{\cdot 1}^T (I_N - R_{F_{\cdot 2}}) F_{\cdot 1} (\theta_1 - \hat{\theta}_1) \le q s^2 F_\alpha(q, N - p)\} \tag{111}$$

where $R_{F_{\cdot 2}} = F_{\cdot 2}(F_{\cdot 2}^T F_{\cdot 2})^{-1} F_{\cdot 2}^T$ and s^2 is an estimate of the error variance σ^2. This region is based on an appropriate subset tangent-plane approximation (See Seber and Wild for details) [59:202-203]. Similar to the original design method, the volume of this region is dependent on

$$|F_{\cdot 1}^T (I_N - R_{F_{\cdot 2}}) F_{\cdot 1}| \tag{112}$$

such that the volume will decrease as the determinant increases. Minimizing the volume of the approximate confidence region leads Hill and Hunter to the following criterion where s denotes subset:

$$\Delta_s = |F_{\cdot 1}^T (I - R_{F_{\cdot 2}}) F_{\cdot 1}| \tag{113}$$

Hill and Hunter go on to explain that the maximization of Δ_s means that the size (as measured by the determinant) of the part of $F_{\cdot 1}$ that is orthogonal to $F_{\cdot 2}$ is maximized [30].

The reduction in complexity results from maximizing the determinant of a smaller matrix. Note that even though a matrix inversion is required, this matrix is not a full $p \times p$ matrix. In other words, the criterion is broken down into smaller, less complicated parts. Press states that LU decomposition, which is used to calculate the determinant on an $N \times N$ matrix, requires $\frac{1}{3}N^3$ matrix operations. For inverting an $N \times N$ matrix, N^3 matrix operations are required [51:37]. Therefore, the original design point determination method requires p^3 matrix operations (p is the total number of parameters) and the parameter subset method discussed here requires $\frac{1}{3}q^3 + r^3$ matrix operations (q is the number of parameters of interest and u is the number of parameters of less interest). Figure 22 compares the operations for the original method and the subset method based on the network architectures used in this chapter. Note that LU decomposition is performed to calculate the determinant for *every* function evaluation required by the maximization routine, so the savings realized by modifying the method is substantial.

Intuition, experimentation and theory all suggest that the lower layer weights in a two layer network are the most important factors in the discrimination process. Intuitively, the lower layer weights perform the initial weighting of the input features thereby sorting the elements of the input vector for further weighting by succeeding layers. Touretzky and Pomerleau state that hidden units should really be called "learned-feature detectors" or "re-representational units," because the activity pattern in the hidden layer is an encoding of what the network thinks are the significant features of the input [64]. Gorman and Sejnowski experimented with sonar targets and drew conclusions from these experiments on the capabilities of the hidden layer. They say that although a hidden unit may be thought of as a feature extractor, the hidden units are also capable of encoding multiple features and even multiple strategies simultaneously [26:88].
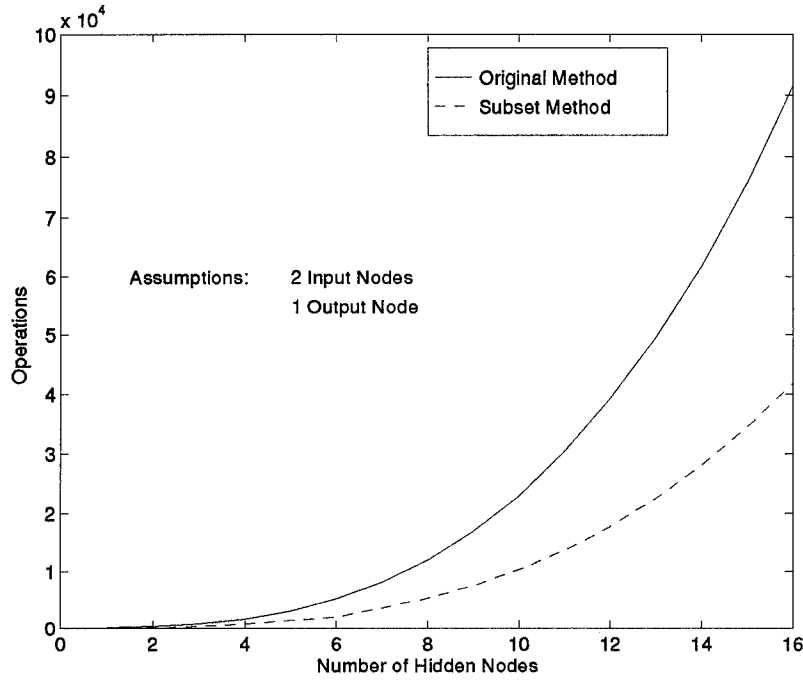
Figure 22. Comparison of Operations for Original Method and Parameter Subset Method

In summary, partitioning the weight vector into parameters of interest and those of less interest simplifies the determination of experimental design points. It seems appropriate to design experiments that emphasize the lower layer weights.

3.4.2.2 An Indicator for the Subset Criterion. In which cases should the lower layer weights be emphasized? Clearly, an indicator is needed to signal that the subset criterion should be used. The original criterion, $|F^T F.|$, can be written

$$
\left| \left[F_{.1} \vdots F_{.2}\right]^T \left[F_{.1} \vdots F_{.2}\right] \right| = \left| \begin{array}{cc} F_{.1}^T F_{.1} & F_{.1}^T F_{.2} \\ F_{.2}^T F_{.1} & F_{.2}^T F_{.2} \end{array} \right|
$$

$$
= \left| F_{.2}^T F_{.2} \right| \left| F_{.1}^T F_{.1} - F_{.1}^T F_{.2} (F_{.2}^T F_{.2})^{-1} F_{.2}^T F_{.1} \right|
$$

$$
= \left| F_{.2}^T F_{.2} \right| \left| F_{.1}^T (I - R_{F_{.2}}) F_{.1} \right| \tag{114}
$$

79

Since this indicator will be used before any design points are chosen, the values of the determinants will be calculated at points from the training data. If the experimenter intends to eventually choose 30 design points, for example, then 30 exemplars chosen randomly from the training set are used to calculate $|F_{.2}^T F_{.2}|$ and $|F_{.1}^T(I - R_{F_{.2}})F_{.1}|$. Replications of this calculation generate observations on $|F_{.2}^T F_{.2}|$ and $|F_{.1}^T(I - R_{F_{.2}})F_{.1}|$.

If a set of design points producing a high value of $|F_{.1}^T(I - R_{F_{.2}})F_{.1}|$ always produces a high value of $|F_{.2}^T F_{.2}|$, it seems reasonable to assume that to maximize $|F^T F|$ it is sufficient to maximize $|F_{.1}^T(I - R_{F_{.2}})F_{.1}|$. An examination of the correlation of $|F_{.2}^T F_{.2}|$ and $|F_{.1}^T(I - R_{F_{.2}})F_{.1}|$ would reveal this type of relationship. Notice, however, that the size of the determinant when measured at sub-optimal (random) locations in the feature space will be very small leading to numerical imprecision. For that reason, a log transformation will be used and the quantities $\log(|F_{.2}^T F_{.2}|)$ and $\log(|F_{.1}^T(I - R_{F_{.2}})F_{.1}|)$ will be examined.

One need only look at the elements of $F_{.1}$ and $F_{.2}$ to see that it is reasonable to assume that $|F_{.2}^T F_{.2}|$ and $|F_{.1}^T(I - R_{F_{.2}})F_{.1}|$ are related. Elements of $F_{.2}$ are $\frac{\partial z}{\partial w_{ij}}$ where w_{ij} is an upper layer weight,

$$\frac{\partial z}{\partial w_{ij}} = z(1 - z)x_i^1 \tag{115}$$

and x_i^1 is the activation of the ith hidden node. Elements of $F_{.1}$ are $\frac{\partial z}{\partial w_{ki}}$ where w_{ki} is a lower layer weight,

$$\frac{\partial z}{\partial w_{ki}} = z(1 - z)w_{i1}^2 x_i^1(1 - x_i^1)x_k \tag{116}$$

where w_{i1}^2 is the weight connecting the ith hidden node to the single output node, and x_k is the kth input. For the single output multilayer perceptron, maximizing the absolute value of the derivative with respect to lower layer weights is nearly equivalent to maximizing the absolute value of the derivative with respect to upper layer weights. The difference is the terms concerning x_i^1 (w_{i1}^2 is a constant). Given that the lower layer derivatives have been maximized, then it follows that the upper layer derivatives have, to some degree, been maximized.

In conclusion, an indicator for when the subset criterion should be used is the estimated correlation of $\log(|F_{\cdot 2}^T F_{\cdot 2}|)$ and $\log(|F_{\cdot 1}^T (I - R_{F_{\cdot 2}}) F_{\cdot 1}|)$ calculated using available training data. If a strong positive correlation is exhibited, then emphasizing the lower layer weights is indicated. It is believed, though cannot yet be proven, that this result holds for all single output multilayer perceptrons.

3.4.2.3 Results—Sample Problem. The two-class problem in Figure 10 was used to test the parameter subset design point method. For this example, 33 design points were used. First, the subset indicator introduced in Section 3.4.2.2 was tested to see if the use of the subset criterion was indicated. Thirty sets of 33 vectors were randomly selected from the training set. Using these 30 observations, the estimated correlation of $\log(|F_{\cdot 2}^T F_{\cdot 2}|)$ and $\log(|F_{\cdot 1}^T (I - R_{F_{\cdot 2}}) F_{\cdot 1}|)$ was calculated as 0.87, indicating a strong relationship between the determinants. The subset criterion was therefore applied.

In Figure 23 the design points obtained when using the entire set of weights are compared with those obtained by using only the weights in the lower layer. The sets of design points differ little in their location in the feature space. Each set of design points was added to the training set and an eight hidden node multilayer perceptron was trained 30 times for each. The resulting classification error on the test set is shown in Figure 24.

Further confirmation that emphasizing the lower layer weights does not significantly degrade the design can be seen in Figure 25. This figure shows the average absolute values of the weight derivatives for the upper and lower weights. (The derivatives were calculated with the initial weight vector.) As stated earlier, one element of the design point criterion is that exemplars are chosen so as to maximize the magnitude of these derivatives. Figure 25 shows that the lower layer weight derivatives are greater in magnitude on average than the upper layer weight derivatives and, therefore, have a greater effect on the the criterion.
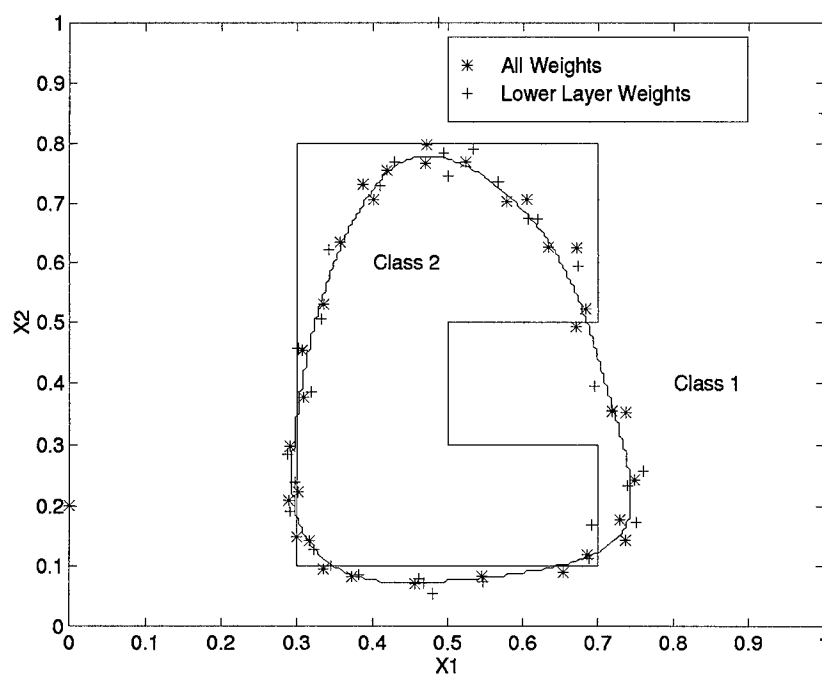
Figure 23. Original Multilayer Perceptron Boundary and Design Points for All Weights and Lower Layer Weights
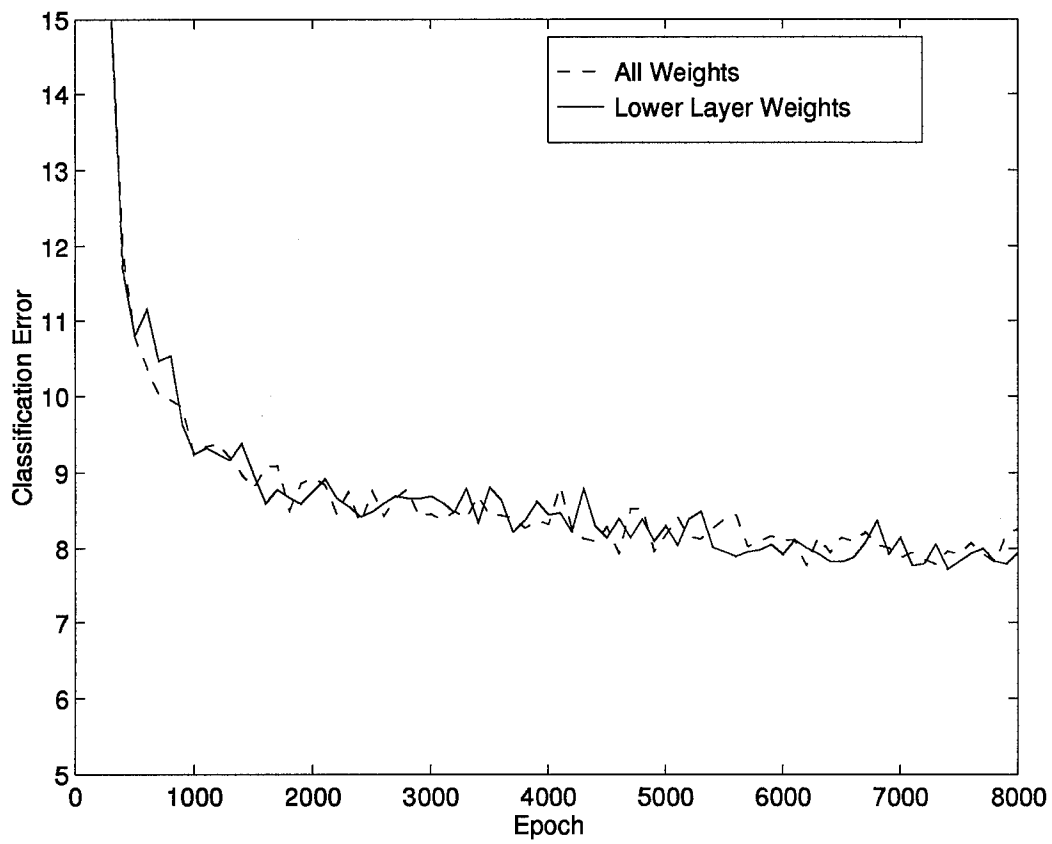
Figure 24. Average Test Set Classification Error—All Weights and Lower Layer Weights(30 Runs, Sampled Every 100 Epochs)
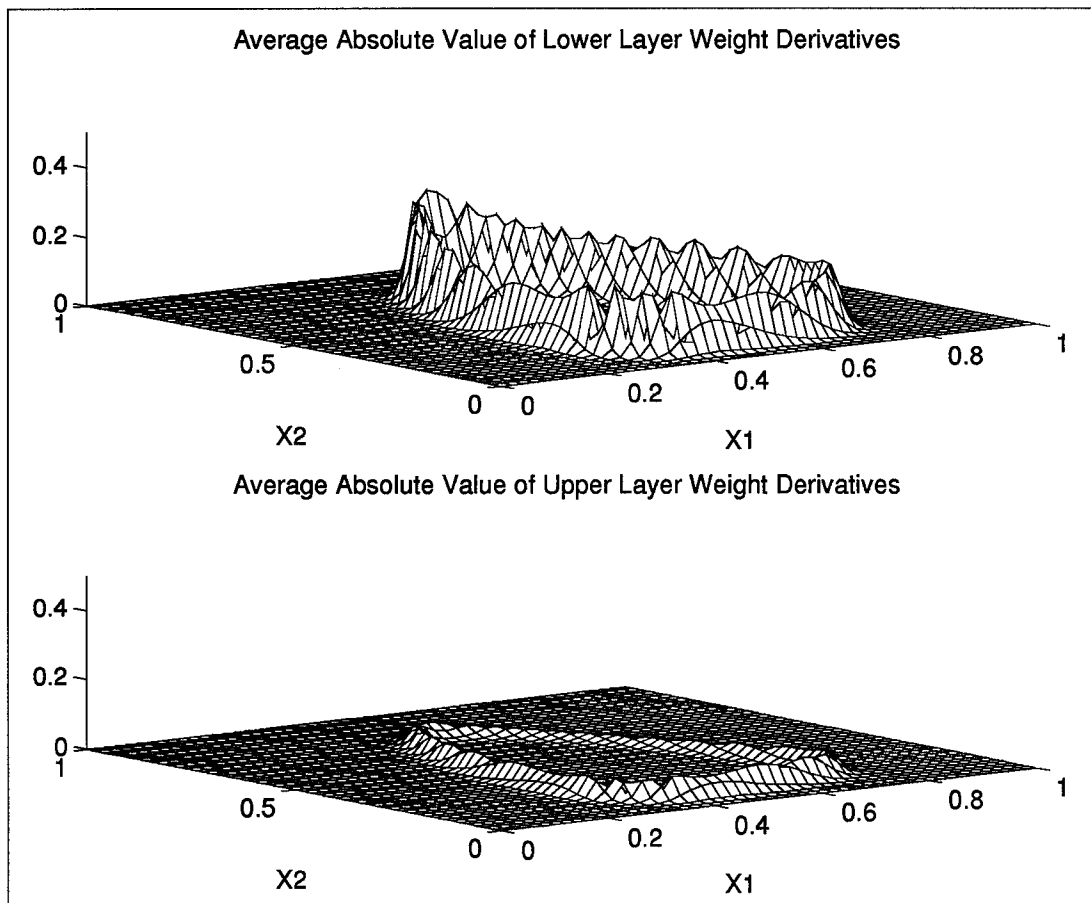
Figure 25. Average Absolute Value of Weight Derivatives

The results here indicate an example in which it was beneficial to emphasize the lower layer weights. For the two-class problems addressed in this section, it appears that the upper layer weights contain little information with regard to designing experiments.

3.5 Sensitivity to Initial Data

3.5.1 Introduction. To this point, it has been assumed that an initial weight vector $\hat{w}$ is available. However, the question of how sensitive the design point determination is to these initial weights has not been established. Box and Lucas observe that

> ... in practical problems it will almost invariably be the case that some information is available, and this will then provide the basis of a first design ... the real effectiveness of the design will depend upon the reliability of the information upon which it is based. [13]

The literature reveals little beyond the statement above. The accuracy or inaccuracy of a given weight vector stems primarily from the training data used to determine the weight vector. If one has a large set of existing data, the initial weights can be determined very efficiently. One might ask, however, that if the initial weights are known with such accuracy, then why are further experiments necessary?

Suppose an initial weight vector has been obtained from an initial set of data. If an improvement in the accuracy of the weights is desired, then the following data manipulation is appropriate:

1. Divide the set S of all existing data into two sets S_T, the training set and S_S, the test set.

2. Using these sets, determine the number of training epochs required, the architecture of the network and other multilayer perceptron settings (example, learning rate). Figure 5

in Chapter I provides a guide. In addition, Steppe presents methods for model and feature selection [61].

3. Fix the multilayer perceptron settings and re-train the network several times.

4. Obtain the weight vector $\hat{w}$ from the run with the lowest test set classification error.

5. Use $\hat{w}$ to determine the design points.

Weiss and Kulikowski discuss the trade-off that exists in choosing how many exemplars to allocate to the training and test sets as in step 1 above. They say "while sufficient test cases are the key to accurate error estimation, adequate training cases in the design of a classifier are also of paramount importance" [67:30]. They go on to state that the usual proportions are approximately a 2/3 (training set), 1/3 (test set) split. Using this partition, however, the error estimate is relatively pessimistic.

When the training is being accomplished in step 3 above, there is no reason that the multilayer perceptron should not be trained to the highest accuracy possible. The only possible danger is "overlearning." Overlearning occurs when the multilayer perceptron memorizes the data in the training set and thereby loses its ability to generalize to unseen data. The inclusion of the test set in the procedure above should remedy this situation.

3.5.2 Test Problem. Figure 26 shows the test problem that will be used to observe the effects of different sample sizes of initial data. This test problem further illustrates the effects of disjoint classes on the design point methodology. Eight hidden nodes were used.

In this problem, the total number of initial data points available to the experimenter was varied between 15 and 480. From this initial data, $\frac{2}{3}$ was allocated to the training set and $\frac{1}{3}$ to the test set. For each train and test set pair, a multilayer perceptron was trained ten times and the run with the lowest classification error on the test set was used. The weight vector from that run was used to identify 33 design points. In addition, in order to adequately judge
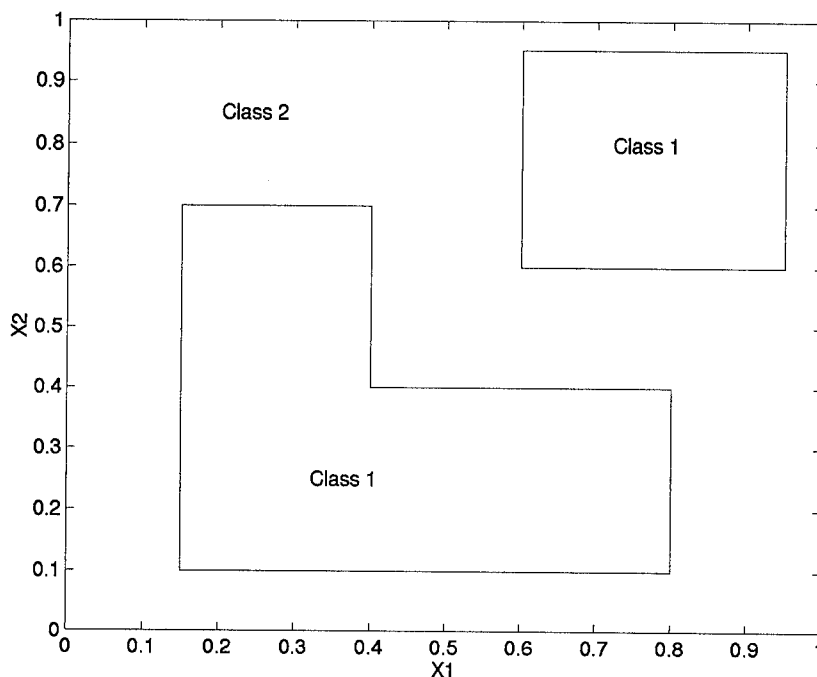
Figure 26. Test Problem—Disjoint Classes

the true accuracy of the classifier, a validation set of 500 vectors was used. Measures on this validation set should be viewed as very good estimates of the true accuracy of the classifier.

Illustrated in Figure 27 are the results of these runs. Several items should be noted from the figure:

- As expected, the more exemplars used to train the classifier, the lower the test set and validation set error rate. To some degree, then, more exemplars yield more accurate weight vectors.

- In almost all cases, the test set error rate was lower than the validation set error rate. This means that an experimenter would be over estimating the accuracy of this classifier if test set error rate was used as the criterion. This fact is not peculiar to this discrimination problem, but is often observed when using multilayer perceptrons.

87

- The case where the total number of samples was 30 illustrates what happens when the multilayer perceptron classifies all exemplars in the same class. Even if an experimenter is certain that two classes exist, he might accept this classifier due to the low test set error rate. The design of experiments method yields an intuitively pleasing result in this case with the design points widely spread over the feature space. Since there are no areas of uncertainty, information is to be gained equally anywhere in the space and the design points reflect this.

So, given that few exemplars yield inaccurate initial weight vectors and many exemplars yield accurate initial weight vectors, how do these accuracies and inaccuracies influence the performance of the final classifier? To investigate this, the design points in each case were added to the training set. Each multilayer perceptron was trained ten times with the augmented training set. The run with the lowest test set classification error was identified in each case and the weights for that run were recorded. The results are shown in Figure 28. Several items should be noted from this figure:

- The "New" column in this figure shows the performance of the multilayer perceptron with the design points included. Again, in almost all cases the validation set error was higher than the test set error.

- The change in the validation set error when the design points are added gets smaller as the number of exemplars gets larger. In other words, the contribution of the design points diminishes as the number of exemplars increases. This is presumably from two causes

  - The network is already fairly accurate without the design points.

  - The design points comprise a smaller and smaller percentage the total data set.

Total Samples	15
Training Set	10
Test Set	5
Test Set Error	0.00
Validation Set Error	34.60

Total Samples	30
Training Set	20
Test Set	10
Test Set Error	10.00
Validation Set Error	34.60

Total Samples	60
Training Set	40
Test Set	20
Test Set Error	10.00
Validation Set Error	21.20

Total Samples	120
Training Set	80
Test Set	40
Test Set Error	5.00
Validation Set Error	21.80

Total Samples	240
Training Set	160
Test Set	80
Test Set Error	7.50
Validation Set Error	10.80

Total Samples	480
Training Set	320
Test Set	160
Test Set Error	3.75
Validation Set Error	5.40

Figure 27. Obtaining Initial Weight Vector

89

- Viewing the figure as a whole, one can see the increasing accuracy of the classifier as the number of exemplars increases. In the 513 sample case, it appears that little is to be gained by selecting further design points.

3.6   Chapter Summary

In this chapter, sampling methods for single output multilayer perceptrons were developed. These statistically based methods select design points for experimentation so as to best estimate the multilayer perceptron weights. The methods presented in this chapter are combined into an overall methodology and applied to a sample problem in Chapter V.

3.6.1   Design of Experiments for Continuous Feature Spaces.

Initially the feature space was assumed to be continuous and Powell's method was used to maximize the Box and Lucas criterion. Experiments performed at the chosen design points and added to the training set exhibited superior accuracy over random design points and over design points chosen is a grid. The fact that the design point methodology outperformed the points chosen in a grid is noteworthy. In practical applications, it is often assumed that gathering data in a grid pattern is optimal. Indeed, volumes of literature exist on selecting factorial designs based on linear or quadratic models—not extremely nonlinear models such as the multilayer perceptron. As shown empirically in this chapter, selecting points in a grid may result in a less accurate classifier.

3.6.2   Design of Experiments for Discrete Feature Spaces.

In practice, many discrimination problems allow experimentation only at discrete points in the feature space. For this reason, a discrete method was developed parallel to the continuous method using a discrete exchange algorithm. Multilayer perceptrons trained with design points chosen from a feasible set by this method exhibited superior accuracy over those trained with randomly selected

	Orig	New
Total Samples	15	48
Training Set	10	43
Test Set	5	5
Test Set Error	0.00	0.00
Validation Set Error	34.60	20.60

	Orig	New
Total Samples	30	63
Training Set	20	53
Test Set	10	10
Test Set Error	10.00	20.00
Validation Set Error	34.60	16.40

	Orig	New
Total Samples	60	93
Training Set	40	73
Test Set	20	20
Test Set Error	10.00	10.00
Validation Set Error	21.20	14.20

	Orig	New
Total Samples	120	153
Training Set	80	113
Test Set	40	40
Test Set Error	5.00	5.00
Validation Set Error	21.80	13.00

	Orig	New
Total Samples	240	273
Training Set	160	193
Test Set	80	80
Test Set Error	7.50	7.50
Validation Set Error	10.80	8.80

	Orig	New
Total Samples	480	513
Training Set	320	353
Test Set	160	160
Test Set Error	3.75	5.00
Validation Set Error	5.40	4.00

Figure 28. After Inclusion of Design Points

91

points. It should be noted that this discrete algorithm could be used for a continuous feature space. All that would be required is the discretization, to some level, of the input features.

3.6.3 Ranking Design Points. Once a set of design points has been chosen, two methods of ranking design points were presented. One method ranks the points according to a simple dot product and the other uses a saliency measure. These ranking methods were shown to result in the selection of the "best" design points.

When contrasted, the ranking methods appear to emphasize different characteristics of the design points. The dot product measure highlights the points where the gradient of the output with respect to the weights is largest. In a well trained multilayer perceptron, the large gradients will appear near the boundary. The saliency ranking highlights the points where the gradient of the output with respect to the inputs is the largest. This measure ranks a point high when its use will cause a large change in the output. It may be that, in certain circumstances, one measure makes more sense than the other.

The D-optimality criterion, in conjunction with the ranking measures, has the potential to be used to choose exemplars from set of points other than design points. Further research may show that exemplars in a training set can be ordered by these methods.

3.6.4 DLF Networks for Design of Experiments. Exploiting the possible functional forms that a multilayer perceptron may take, a Direct Linear Feedthrough (DLF) network was introduced. The DLF was appealing for several reasons:

1. From a training point of view, if there are linearities present in the discrimination problem, a DLF network allows one to develop a classifier with fewer hidden nodes and fewer weights.

2. In a design of experiments setting, the parameters associated with the linear part of the model may be ignored.

3. The elements of the design point criterion associated with the linear parts of the DLF network are very simple.

The result is a reduction in the complexity of the design point criterion.

3.6.5 Subsets of Parameters. A second approach to reducing the complexity of the design point criterion presented here involved developing a new criterion which emphasized the lower layer weights in the multilayer perceptron. It appears that, for the single output multilayer perceptron, it is sufficient to consider the lower layer weights when selecting design points. This simplification considerably reduces the number of calculations required.

3.6.6 Sensitivity to Initial Data. The test problem observed for investigating the effects of the initial weight vector $\hat{w}$ confirmed many previously known results on multilayer perceptrons. First, is was demonstrated that the more exemplars used to train the classifier, the more accurate the result—an intuitive result. Second, it was shown that test set error rates should only be used as indicators and can vary greatly from the population error rate. In terms of design point selection, it was shown that as the number of initial exemplars increases, the benefit gained from the design points decreases. Finally, an example was given for a multilayer perceptron which erroneously classifies the entire feature space as one class. Design points were placed in a seemingly random pattern over the feature space, demonstrating the robustness of the method.

IV. Design of Experiments for Multiple Output Multilayer Perceptrons

4.1 Introduction

To this point, only single output multilayer perceptrons have been addressed. The introduction of multiple output multilayer perceptrons greatly increases the complexity of the experimental design point methodology. As in the single output case, the theory and results in this chapter are based on a multilayer perceptron with a single layer of middle nodes and sigmoidal activations at the middle and output nodes. Unlike the two-class case, multi-class discrimination problems require more than one multilayer perceptron output. The convention is to use one output for each class with the desired output coded as "1" for the correct class and "0" for all other classes. A vector is, therefore, classified according to the node with the greatest output.

Networks with multiple outputs correspond to multivariate nonlinear regression models. Compared with univariate research, a limited amount of work has been done in the area of design of experiments for nonlinear models with multivariate response.

In this chapter, the D-optimality criterion for multi-response models is revisited. Next, a criterion is developed for a discrete feature space. Results are then presented for a four-class discrimination problem. Finally, a method of reducing the complexity of the design point criterion is formulated and demonstrated.

4.1.1 Multi-Response D-Optimality for Multilayer Perceptrons.
As stated in Equation 64, the criterion used for multi-response models is

$$|D| = \left| \sum_{i=1}^{r} \sum_{j=1}^{r} \sigma^{ij} F_{.i}^{T} F_{.j} \right| \tag{117}$$

where σ^{ij} are the elements of the known inverted variance-covariance matrix and $|D|$ is to be maximized. This criterion can be estimated by

$$|\hat{D}| = \left| \sum_{i=1}^{r} \sum_{j=1}^{r} \hat{v}^{ij} F_{\cdot i}^{T} F_{\cdot j} \right| \qquad (118)$$

where $\hat{v}^{ij}$ is the ijth element of the inverse of the estimated variance-covariance matrix. (See Section 2.4) All available data vectors $(\mathbf{x}_s, s = 1, \ldots, N)$ are used to estimate $\hat{v}^{ij}$ with

$$\hat{v}_{ij} = \frac{1}{N} \sum_{s=1}^{N} [y_{is} - f_i(\mathbf{x}_s; \boldsymbol{\theta})][y_{js} - f_j(\mathbf{x}_s; \boldsymbol{\theta})] \qquad (119)$$

and $\{\hat{v}^{ij}\} = \{\hat{v}_{ij}\}^{-1}$, $i, j = 1, 2, \ldots, r$. The observations required for calculating these estimates are readily available during the training of the multilayer perceptron.

Viewed simply, the multi-response D-optimal criterion is a weighted sum of single response D-optimal criterion taken over all possible response pairs. Observe that when $r = 1$, the criterion in Equation 118 is identical to the one given by Box and Lucas (used in Chapter II) [13]. One can gain further insight by supposing there are only two responses, y_1 and y_2, and y_1 can be measured much more accurately than y_2. In this case, $\sigma_1^2 << \sigma_2^2$. Letting ρ be the correlation between y_1 and y_2, then

$$\Sigma^{-1} = \begin{bmatrix} \dfrac{1}{\sigma_1^2(1-\rho^2)} & \dfrac{-\rho}{\sigma_1\sigma_2((1-\rho^2)} \\[3mm] \dfrac{-\rho}{\sigma_1\sigma_2((1-\rho^2)} & \dfrac{1}{\sigma_2^2(1-\rho^2)} \end{bmatrix} \qquad (120)$$

and therefore, $\sigma^{11} >> \sigma^{22}$. The criterion gives greater weight to $F_{\cdot 1}^{T} F_{\cdot 1}$ than to $F_{\cdot 2}^{T} F_{\cdot 2}$ and thus emphasizes the selection of design points appropriate for y_1 [20].

In the single response case, when the feature space is discrete (or assumed to be discrete), design points were chosen so as to maximize

$$\hat{\mathbf{f}}_{\cdot}^{T}(\mathbf{x})(\hat{F}_{\cdot}^{T}\hat{F}_{\cdot})^{-1}\hat{\mathbf{f}}_{\cdot}(\mathbf{x}) \qquad (121)$$

95

Choosing points in this way ensured maximization of $|\hat{F}.^T\hat{F}.|$ for the current set of design points. In the multi-response case, the criterion in Equation 118 must be maximized.

Given an initial set of N design points, each $F._i$ $(i = 1, \ldots, r)$ and $|D|$ can be calculated. If an additional vector $\mathbf{x}_a$ is added to the design, the criterion becomes

$$
\begin{aligned}
|\tilde{D}| &= \left| \sum_{i=1}^{r} \sum_{j=1}^{r} \hat{v}^{ij} \begin{bmatrix} F._i \\ \mathbf{f}._i^T(\mathbf{x}_a) \end{bmatrix}^T \begin{bmatrix} F._j \\ \mathbf{f}._j^T(\mathbf{x}_a) \end{bmatrix} \right| \\
&= \left| \sum_{i=1}^{r} \sum_{j=1}^{r} \hat{v}^{ij} \begin{bmatrix} F._i^T & \mathbf{f}._i(\mathbf{x}_a) \end{bmatrix} \begin{bmatrix} F._j \\ \mathbf{f}._j^T(\mathbf{x}_a) \end{bmatrix} \right| \\
&= \left| \sum_{i=1}^{r} \sum_{j=1}^{r} \hat{v}^{ij} F._i^T F._j + \sum_{i=1}^{r} \sum_{j=1}^{r} \hat{v}^{ij} \mathbf{f}._i(\mathbf{x}_a) \mathbf{f}._j^T(\mathbf{x}_a) \right|
\end{aligned}
\tag{122}
$$

where $F._i$ is $N \times p$ and $\mathbf{f}._i$ is $p \times 1$.

Let $\mathbf{f}.(\mathbf{x}_a) = [\mathbf{f}._1(\mathbf{x}_a), \mathbf{f}._2(\mathbf{x}_a), \ldots, \mathbf{f}._r(\mathbf{x}_a)]$, $\hat{\Sigma}^{-1} = \{\hat{v}^{ij}\}$,

$$
F. = \begin{bmatrix} F._1 \\ F._2 \\ \vdots \\ F._r \end{bmatrix}
\tag{123}
$$

and define the Kronecker product of A $(m \times m)$ and B $(n \times n)$ as [59]

$$
A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mm}B \end{bmatrix}
\tag{124}
$$

Then,

$$
\sum_{i=1}^{r} \sum_{j=1}^{r} \hat{v}^{ij} F._i^T F._j = F.^T (\Sigma^{-1} \otimes I_N) F.
\tag{125}
$$

96

and

$$\sum_{i=1}^{r}\sum_{j=1}^{r}\hat{v}^{ij}\mathbf{f}_{\cdot i}(\mathbf{x}_a)\mathbf{f}_{\cdot j}^{T}(\mathbf{x}_a) = \mathbf{f}_{\cdot}(\mathbf{x}_a)\hat{\Sigma}^{-1}\mathbf{f}_{\cdot}^{T}(\mathbf{x}_a) \tag{126}$$

so that

$$\begin{aligned}|\tilde{D}| &= \left|F_{\cdot}^{T}(\Sigma^{-1}\otimes I_N)F_{\cdot} + \mathbf{f}_{\cdot}(\mathbf{x}_a)\hat{\Sigma}^{-1}\mathbf{f}_{\cdot}^{T}(\mathbf{x}_a)\right| \\ &= \left|D + \mathbf{f}_{\cdot}(\mathbf{x}_a)\hat{\Sigma}^{-1}\mathbf{f}_{\cdot}^{T}(\mathbf{x}_a)\right|\end{aligned} \tag{127}$$

The matrix $\hat{\Sigma}^{-1}$ is symmetric, so by the spectral decomposition theorem, $\hat{\Sigma}^{-1}$ can be written as

$$\hat{\Sigma}^{-1} = P\Upsilon P^{T} \tag{128}$$

where Υ is a diagonal matrix of eigenvalues of $\hat{\Sigma}^{-1}$ and P is an orthogonal matrix whose columns are the normalized eigenvectors associated with the diagonal entries of Υ [19]. Then, writing Υ as $\Upsilon^{\frac{1}{2}}(\Upsilon^{\frac{1}{2}})^{T}$,

$$|\tilde{D}| = \left|D + \mathbf{f}_{\cdot}(\mathbf{x}_a)P\Upsilon^{\frac{1}{2}}(\Upsilon^{\frac{1}{2}})^{T}P^{T}\mathbf{f}_{\cdot}^{T}(\mathbf{x}_a)\right| \tag{129}$$

Let $A = \mathbf{f}_{\cdot}(\mathbf{x}_a)P\Upsilon^{\frac{1}{2}}$. Then

$$|\tilde{D}| = \left|D + AA^{T}\right| \tag{130}$$

Using the identity [48:210],

$$\begin{vmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{vmatrix} = |B_{11}||B_{22} + B_{21}B_{11}^{-1}B_{12}| = |B_{22}||B_{11} + B_{12}B_{22}^{-1}B_{21}| \tag{131}$$

97

implies $|D + AA^T| = |D||I + A^T D^{-1} A|$. Therefore, the multi-response discrete design point algorithm examines

$$\left| I + \left(\mathbf{f}.(\mathbf{x}_a) P \Upsilon^{\frac{1}{2}} \right)^T D^{-1} \left(\mathbf{f}.(\mathbf{x}_a) P \Upsilon^{\frac{1}{2}} \right) \right| \tag{132}$$

and chooses (for inclusion in the design point set) the point $\mathbf{x}_a$ that yields the largest value of this quantity. The above derivation is also applicable in a sequential design of experiments approach. For the single response case, see Equation 51.

4.1.2 Notation. All notation developed in Chapter III, Section 3.1.2 can still be applied. In addition, the matrix of first partials will now be subscripted with the index of the output node, i.e.,

$$F_{\cdot j}(\mathbf{w}) = \left\{ \frac{\partial z_j^s}{\partial w_t} \right\} \qquad s = 1, \ldots, N; \quad j = 1, \ldots, r; \quad t = 1, \ldots, p \tag{133}$$

These derivatives differ from the single response case only slightly, that is

$$w_t = \begin{cases} w^1_{\lceil \frac{t}{m} \rceil, t - m(\lceil \frac{t}{m} \rceil - 1)} & \text{for } t \le (n+1)m \\[2ex] w^2_{\lceil \frac{t}{(n+1)m+r} \rceil, t - r(\lceil \frac{t}{(n+1)m+r} \rceil - 1) - (n+1)m} & \text{for } (n+1)m < t \le (n+1)m + (m+1)r \end{cases} \tag{134}$$

and $\lceil \alpha \rceil$ is the smallest integer greater than α. Then, for lower layer weights,

$$\frac{\partial z_j^s}{\partial w_{ki}^1} = z_j^s (1 - z_j^s) w_{ij}^2 x_i^{1s} (1 - x_i^{1s}) x_k^s \tag{135}$$

where x_i^{1s} is the activation of the ith middle node given the input vector s and x_k^s is the kth element of the input vector s. Similarly for upper layer weights,

$$\frac{\partial z_j^s}{\partial w_{ij}^2} = z_j^s (1 - z_j^s) x_i^{1s} \tag{136}$$

98

The estimated variance-covariance matrix for the responses based on N exemplars in the notation of multilayer perceptrons is

$$\hat{v}_{ij} = \frac{1}{N} \sum_{s=1}^{N} [d_i^s - z_i^s][d_j^s - z_j^s] \tag{137}$$

where d_i^s is the desired output for the ith output node given the sth exemplar and z_i^s is the actual output of the ith output node for the sth exemplar.

4.2 Results

To demonstrate the multi-response criterion, a two-dimensional, four-class discrimination problem was used. A multilayer perceptron with 10 hidden nodes, 50 training vectors and 500 test vectors was trained 30 times and the weight vector was chosen from the run with the lowest classification error on the test set. Figure 29 shows the discrimination problem and the original multilayer perceptron boundaries.

The feature space was assumed to be continuous and Powell's method was used to maximize the determinant criterion and choose 30 design points. Figure 30 shows the resulting design points and the boundary formed by the trained multilayer perceptron. Note that, similar to the two-class results, the design points are near the boundaries of the trained multilayer perceptron. Calculation of these design points was a lengthy process requiring 41.54 hours of system time on a Sun Sparc station 5. Figure 31 shows a second iteration of the design point method (again choosing 30 points). Finally, Figure 32 shows the average test set classification error over 30 runs for the original multilayer perceptron and for Iterations 1 and 2 as compared to randomly chosen design points. The final average test set classification error for the results shown in Figure 32 are given in Table 4. The difference between the mean classification error for the design points and the random points was statistically significant ($\alpha = 0.05$) for both iterations.

99
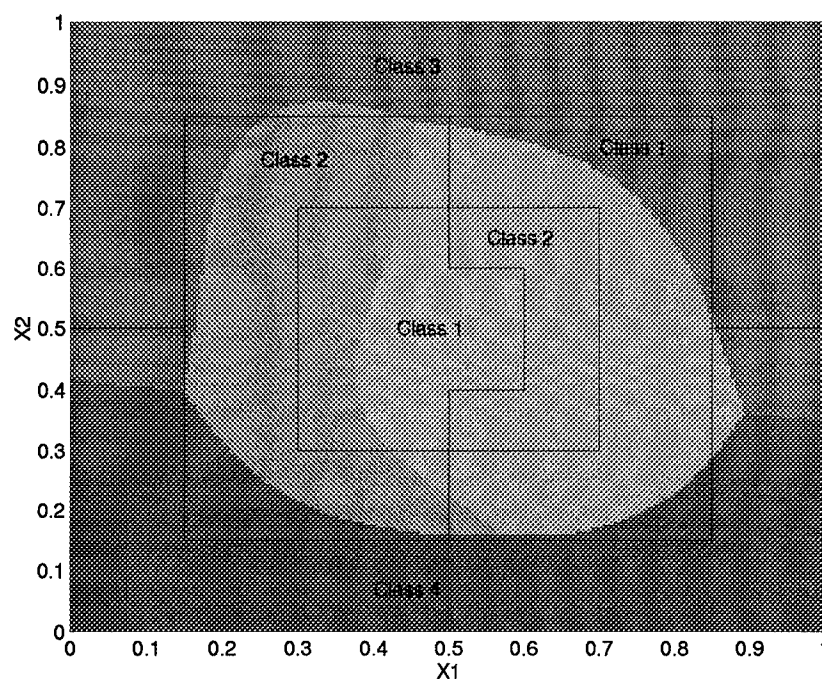
Figure 29. Multiple Output Discrimination Problem and Original Multilayer Perceptron Boundary

Table 4. Multiple Output Discrimination Problem—Final Average Test Set Classification Errors

	Set Added to Training Set	Average Test Set Classification Error
	None (Original Data)	26.85
Iteration 1	Design Point Method	21.78
	Random Points	23.42
Iteration 2	Design Point Method	18.86
	Random Points	20.04
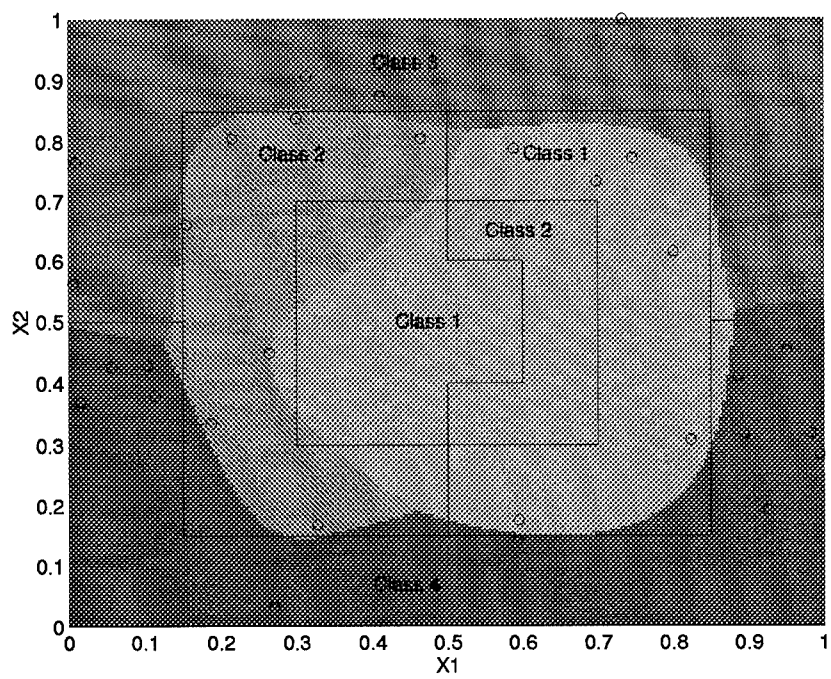
Figure 30. Multiple Output Discrimination Problem—Design Points and Resulting Boundary (Iteration 1)
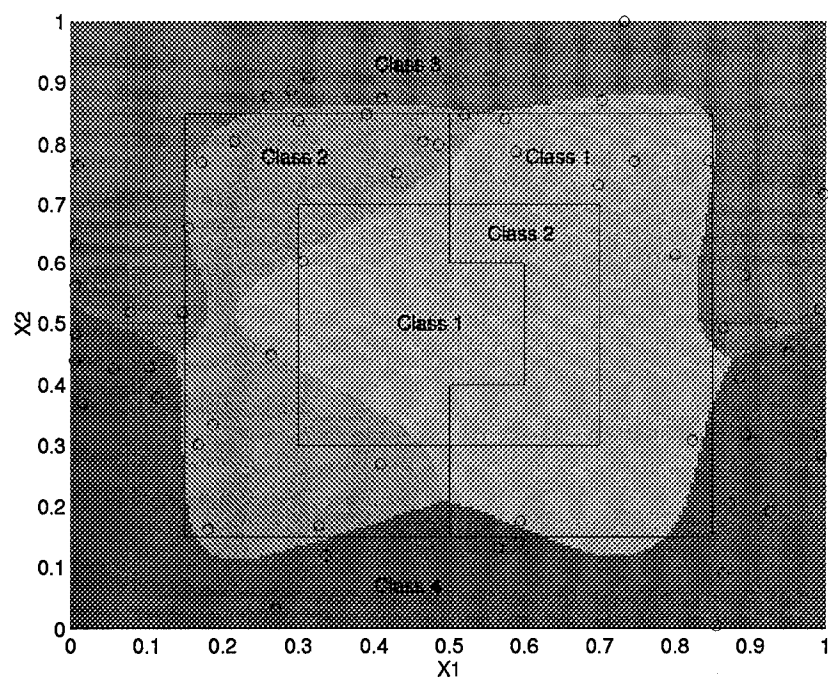
Figure 31. Multiple Output Discrimination Problem—Design Points and Resulting Boundary (Iteration 2)
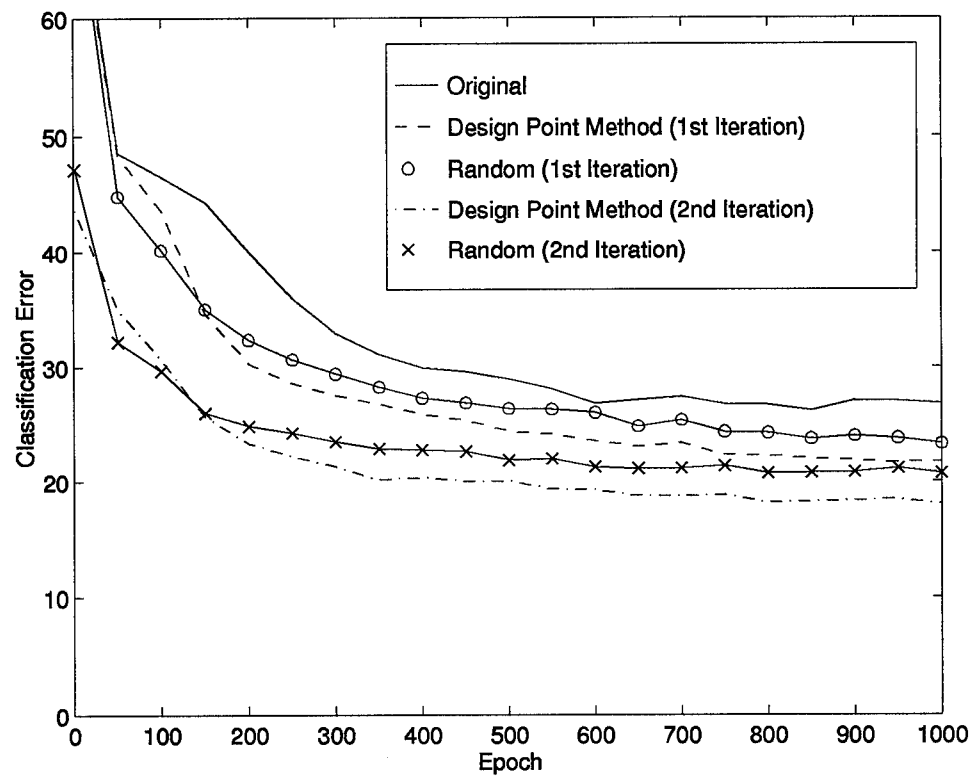
Figure 32. Multiple Output Discrimination Problem—Average Test Set Classification Error

The saliency ranking measure ($\mathcal{M}_2$) introduced in Section 3.3.3 is directly applicable to multi-class discrimination problems. This measure was defined as

$$\mathcal{M}_2(\mathbf{x}) = \sum_{j=1}^{r} \sum_{k=1}^{n} \left| z_j(1 - z_j) \sum_{i=1}^{m} w_{ij}^2 x_i^1 (1 - x_i^1) w_{ki}^1 \right| \tag{138}$$

where $\mathbf{x}$ is the design point under consideration and current weight estimates $\hat{\mathbf{w}}$ are used. Design points will be ordered according to this measure with large values corresponding to "good" design points. Figures 33 and 34 show the design points and the value of $\mathcal{M}_2$ for each design point. Figure 33 corresponds to the data points found in the first iteration above and Figure 34 corresponds to the second iteration. The highest ranked design points in the first iteration are clustered near $x_1=0$ and $x_2=0.5$, while the points from the second iteration are clustered around $x_1=0.5$ and $x_2=0.8$. It appears that each iteration emphasized a different area of the feature space resulting from different initial weight vectors.

4.3   Reducing the Complexity of Design Point Determination

The design point criterion as it currently stands is simple in form. However, it requires a huge number of calculations. To calculate the design point criterion for a single candidate set of design points requires r^2 evaluations of the form $\hat{v}^{ij} F_{\cdot i}^T F_{\cdot j}$. Each matrix $F_{\cdot i}$ requires Np evaluations of the derivative $\frac{\partial z_i}{\partial \mathbf{w}}$. In turn, these derivatives require evaluation of the multilayer perceptron at each of N points. All of these calculations are required for a *single* evaluation of a *single* term in the design point criterion. The Powell algorithm may require thousands of evaluations of the criterion. Needless to say, simplification of the criterion is a necessary step toward practical implementation.

The multi-response design point criterion (Equation 118) could be simplified if the responses were uncorrelated, i.e., if $\hat{\Sigma} = P$, a diagonal matrix. Then, rather than r^2 terms of
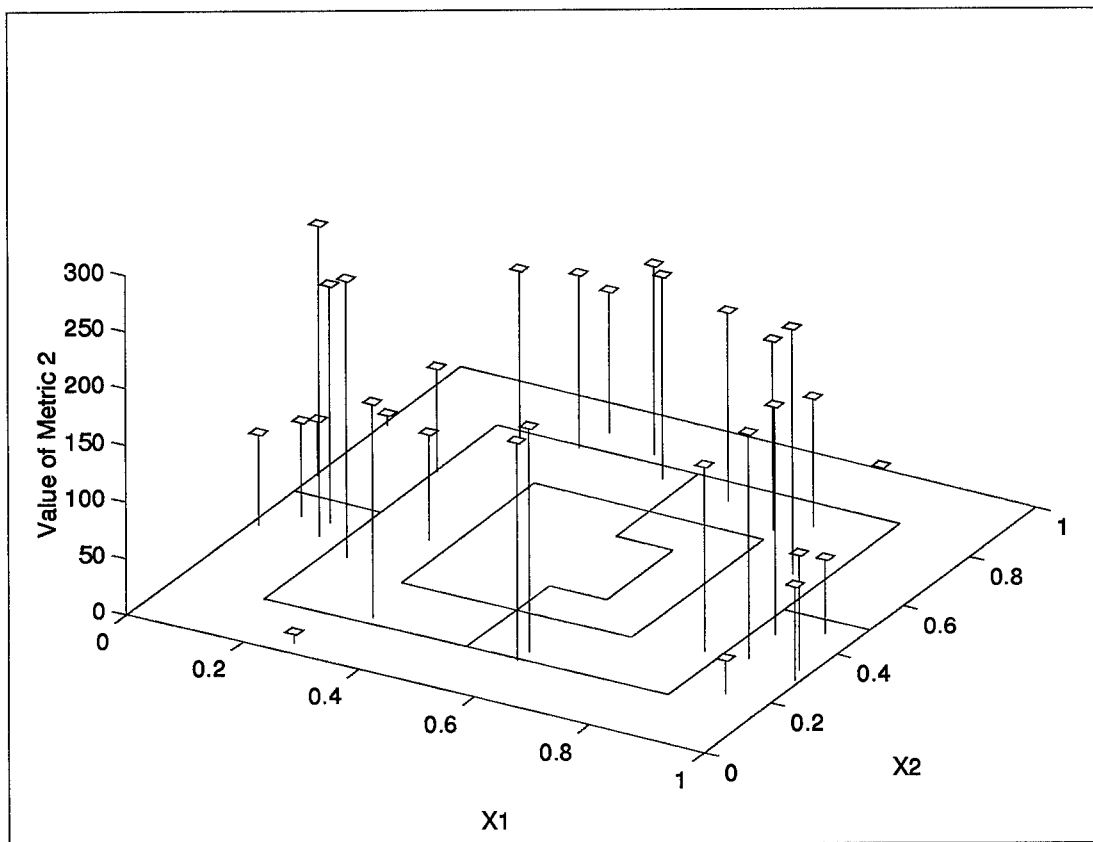
Figure 33. Design Points and $\mathcal{M}_2$—Iteration 1
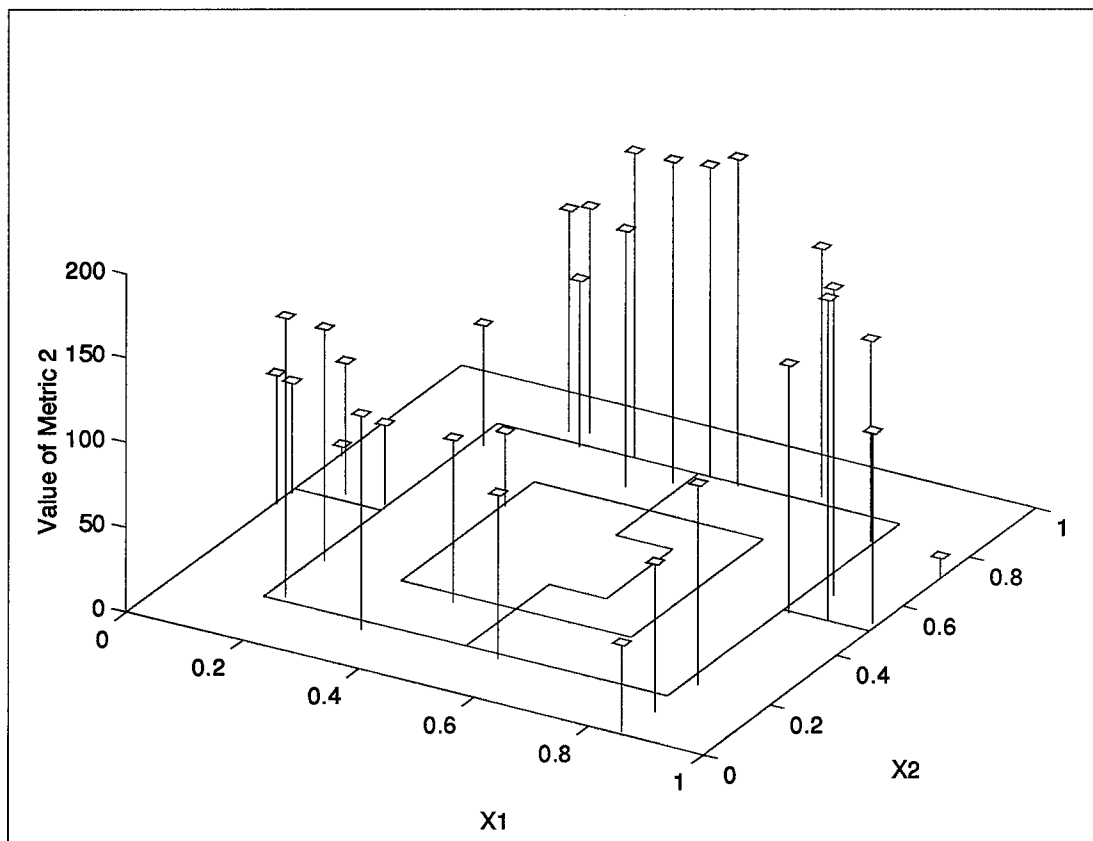
Figure 34. Design Points and $\mathcal{M}_2$—Iteration 2

the form $\hat{v}^{ij} F_{.i}^T F_{.j}$ there would be r terms of that form. Considering the fact that each term includes the multiplication of two $N \times p$ matrices, the savings is significant.

One way to accomplish this goal is to "pre-set" the desired outputs in a manner that will produce uncorrelated actual outputs. As a multilayer perceptron trains, its actual outputs move closer and closer to the desired outputs. Therefore, it seems reasonable that the desired outputs can be used as estimates for the actual outputs. In addition, these desired outputs can be used to surmise the form of the variance-covariance matrix for the actual outputs. This can be shown as follows.

In general, the covariance of y_i and y_j is defined as

$$\text{cov}(y_i, y_j) = E\left\{ [y_i - E[y_i]] \, [y_j - E[y_j]] \right\} \tag{139}$$

The covariance can also be expressed as

$$\text{cov}(y_i, y_j) = E\left[y_i y_j\right] - E\left[y_i\right] E\left[y_j\right] \tag{140}$$

An estimate of the elements of the variance-covariance matrix is

$$\tilde{v}_{ij} = \frac{1}{N} \sum_{s=1}^{N} y_{is} y_{js} - \mu_i \mu_j \tag{141}$$

where N is the number of observations, y_{is} is the value of the sth observation on the ith element of y and $\mu_i = \frac{1}{N} \sum_{s=1}^{N} y_{is}$.

Let d_{is}^k be the desired value of output node i for the sth exemplar given that this exemplar is in the kth class. Then, in multilayer perceptron notation, an estimate of the variance-covariance matrix of the desired outputs is given by

$$\tilde{v}_{ij} = \frac{1}{N} \sum_{s=1}^{N} d_{is} d_{js} - \mu_i \mu_j \qquad i, j = 1, \ldots, r \tag{142}$$

with the $\cdot$ meaning "without reference to" and $\mu_i = \frac{1}{N} \sum_{s=1}^{N} d_{is}$. Since actual outputs approach desired outputs during training, it seems reasonable that $\tilde{v}^{ij}$ be used as an estimate of the variance-covariance matrix of actual outputs. Since d_{is}^k will be the same for all exemplars in class k, $\tilde{v}_{ij}$ can be rewritten as

$$\tilde{v}_{ij} = \frac{1}{r} \sum_{k=1}^{r} d_{i\cdot}^k d_{j\cdot}^k - \mu_i \mu_j \qquad i,j = 1, \ldots, r \tag{143}$$

(This equation assumes an equal number of exemplars from each class.)

In order to arrive at a diagonal estimated variance-covariance matrix, one must choose desired outputs so that the following holds:

1. $\tilde{v}_{ij} = 0$ for all $i \neq j$.

2. The class representations are unique. Defining the distance, λ, between class representations i and j as:

$$\lambda_{ij} = \left\{ \sum_{k=1}^{r} (d_{k\cdot}^i - d_{k\cdot}^j)^2 \right\}^{\frac{1}{2}} \qquad i,j = 1, \ldots, r \tag{144}$$

This distance must be greater than zero.

3. The desired outputs should be such that

$$d_{i\cdot}^k \in \{0, 1\} \qquad i, k = 1, \ldots, r \tag{145}$$

and must be such that

$$d_{i\cdot}^k \in [0, 1] \qquad i, k = 1, \ldots, r \tag{146}$$

Even when the above constraints are satisfied, one is not guaranteed that the *actual* outputs of the trained network will be uncorrelated. The resulting outputs should be approxi-

mately uncorrelated if the network is sufficiently well-trained and if the number of exemplars in each class is approximately equal.

4.3.1 Finding Appropriate Desired Outputs.

4.3.1 Finding Appropriate Desired Outputs. The next question to be answered is how one actually finds desired outputs to satisfy the above constraints. The definition of a special class of matrices called *Hadamard* matrices is now required:

Definition 2 *The* **Hadamard matrix** *is a square array whose elements are only +1 and -1 and whose rows and columns are orthogonal to one another. A symmetrical Hadamard matrix with the first column containing +1's is known as the* **normal form** *for the Hadamard matrix. The lowest order Hadamard matrix is of order two:*

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{147}$$

Higher order matrices restricted to having powers of two, can be obtained from the recursive relationship

$$H_N = H_{\frac{N}{2}} \otimes H_2 \tag{148}$$

where N is a power of 2 and $\otimes$ denotes the Kronecker product defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mm}B \end{bmatrix} \tag{149}$$

[4:28-29]

Table 5. Desired Outputs for Four Class Discrimination

	Class 1	Class 2	Class 3	Class 4
Output 1	1	1	1	1
Output 2	1	0	1	0
Output 3	1	1	0	0
Output 4	1	0	0	1

Hadamard matrices can be used to easily determine desired outputs for discrimination problems when the number of classes is equal to a power of two. For example, for four classes

$$H_4 = H_2 \otimes H_2$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{150}$$

$$\tag{151}$$

As stated earlier, desired outputs should be 0 or 1. Substituting 0's for -1's in H_4 results in desired outputs that satisfy the three constraints listed above. These vectors are listed in Table 5. Notice that Output 1 is the same (1) for each class and makes no contribution to classification. Therefore, this output will be removed leaving three outputs for the classification of four classes. The idealized form of the variance-covariance matrix and distance matrix for the three desired outputs is

$$\tilde{\Sigma} = \begin{bmatrix} 0.3333 & 0 & 0 \\ 0 & 0.3333 & 0 \\ 0 & 0 & 0.3333 \end{bmatrix} \tag{152}$$

110

Table 6. Choosing Desired Outputs for Five Class Discrimination

	Class 1	Class 2	Class 3	Class 4	Class 5
Output 1	1	1	1	1	1
Output 2	1	0	1	0	d_{25}
Output 3	1	1	0	0	d_{35}
Output 4	1	0	0	1	d_{45}
Output 5	1	1	1	1	0

and the distance matrix is

$$\mathcal{L} = \begin{bmatrix} * & \sqrt{2} & \sqrt{2} & \sqrt{2} \\ \sqrt{2} & * & \sqrt{2} & \sqrt{2} \\ \sqrt{2} & \sqrt{2} & * & \sqrt{2} \\ \sqrt{2} & \sqrt{2} & \sqrt{2} & * \end{bmatrix} \tag{153}$$

Training to these desired outputs should result in a variance-covariance matrix that is approximately diagonal.

By changing the desired outputs, the method of classification of input vectors must also change. Previously, an input vector was classified according to the greatest output node. With this new scheme, an input vector will be classified according to the closest vector of desired outputs in a Euclidean distance sense.

The question remains as to how to select desired outputs when the number of classes is not a power of two. For an r-class problem, this can be accomplished by simply augmenting the desired outputs for an $(r - 1)$-class problem. For five classes, begin with the desired outputs for four classes. Augment the first row with an additional "1." Augment an additional row of 1's and a single 0 at d_5^5.. (See Table 6)

Table 7. Desired Outputs for Five Class Discrimination

	Class 1	Class 2	Class 3	Class 4	Class 5
Output 1	1	1	1	1	1
Output 2	1	0	1	0	0.5
Output 3	1	1	0	0	0.5
Output 4	1	0	0	1	0.5
Output 5	1	1	1	1	0

Now, in order to maintain $\tilde{v}_{ij} = 0$ for $i \neq j$, it must be true that $\tilde{v}_{25}, \tilde{v}_{35}$ and $\tilde{v}_{45}$ equal zero. This yields

$$\sum_{k=1}^{r} d_{2.}^{k}.d_{5.}^{k} = 5\mu_2\mu_5$$

$$\sum_{k=1}^{r} d_{3.}^{k}.d_{5.}^{k} = 5\mu_3\mu_5$$

$$\sum_{k=1}^{r} d_{4.}^{k}.d_{5.}^{k} = 5\mu_4\mu_5 \tag{154}$$

Substituting in the known desired outputs results in

$$d_{2.}^{5} = 0.5 \qquad d_{3.}^{5} = 0.5 \qquad d_{4.}^{5} = 0.5 \tag{155}$$

The desired outputs in Table 7 satisfy the criteria for desired outputs in a five class discrimination problem. Again, Output 1 is the same for all classes and will be deleted. The idealized form of the variance-covariance matrix is

$$\tilde{\Sigma} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0.20 \end{bmatrix} \tag{156}$$

112

Table 8. Desired Outputs for Six Class Discrimination

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Output 1	1	1	1	1	1	1
Output 2	1	0	1	0	0.5	0.5
Output 3	1	1	0	0	0.5	0.5
Output 4	1	0	0	1	0.5	0.5
Output 5	1	1	1	1	0	0.8
Output 6	1	1	1	1	1	0

and the distance matrix is

$$
\mathcal{L} = \begin{bmatrix}
* & \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{1.75} \\
\sqrt{2} & * & \sqrt{2} & \sqrt{2} & \sqrt{1.75} \\
\sqrt{2} & \sqrt{2} & * & \sqrt{2} & \sqrt{1.75} \\
\sqrt{2} & \sqrt{2} & \sqrt{2} & * & \sqrt{1.75} \\
\sqrt{1.75} & \sqrt{1.75} & \sqrt{1.75} & \sqrt{1.75} & *
\end{bmatrix}
\tag{157}
$$

Note that, to obtain a solution, it was necessary to relax the constraint that the values of the desired output be 0 or 1 and use 0.5. The distance matrix indicates that multilayer perceptron training should not suffer from this "three-level" coding. The separation between classes is still fairly large.

Similarly, six and seven class desired outputs can be obtained by augmentation. Given the five-class solution, which is known to satisfy the constraints, augment a sixth row and column to obtain the desired outputs in Table 8. Then given the six-class solution, augment a seventh row and column to obtain the desired outputs in Table 9. Both the six and seven class desired outputs given should result in approximately diagonal variance-covariance matrices. For an eight-class problem, the H_8 Hadamard matrix (transformed to 0's and 1's) can be used with augmentation performed for the nine-class problem, and so on. In each case, the

113

Table 9. Desired Outputs for Seven Class Discrimination

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Output 1	1	1	1	1	1	1	1
Output 2	1	0	1	0	0.5	0.5	0.5
Output 3	1	1	0	0	0.5	0.5	0.5
Output 4	1	0	0	1	0.5	0.5	0.5
Output 5	1	1	1	1	0	0.8	0.8
Output 6	1	1	1	1	1	0	0.83333
Output 7	1	1	1	1	1	1	0

multilayer perceptron is reduced by a single output since the outputs for that node are equal. (Here, Output 1 was the deleted output.)

4.3.2 Simplified Design Point Criterion. When the desired outputs are adjusted as described in the section above, it may be that the estimate of the variance-covariance matrix based on *actual* outputs is not approximately diagonal. The desired and actual outputs will always differ by some amount. One cannot expect the actual outputs to behave exactly as the desired outputs.

If the form is approximately diagonal, the design point criterion becomes

$$|D| = \left| \sum_{i=1}^{r} \frac{1}{\tilde{v}_{ii}} F_{\cdot i}^{T} F_{\cdot i} \right| \tag{158}$$

significantly reducing the number of calculations. Design points are then chosen based on this reduced criterion. Figure 35 shows the simplified design point method.

4.3.3 Results. Initially, the simple test problem shown in Figure 36 was used to verify the method. Success in obtaining uncorrelated outputs is dependent on the degree to which the multilayer perceptron was trained. To produce a very well trained network, 1000 training exemplars were used to train a multilayer perceptron with six hidden nodes for 5000 epochs.

114
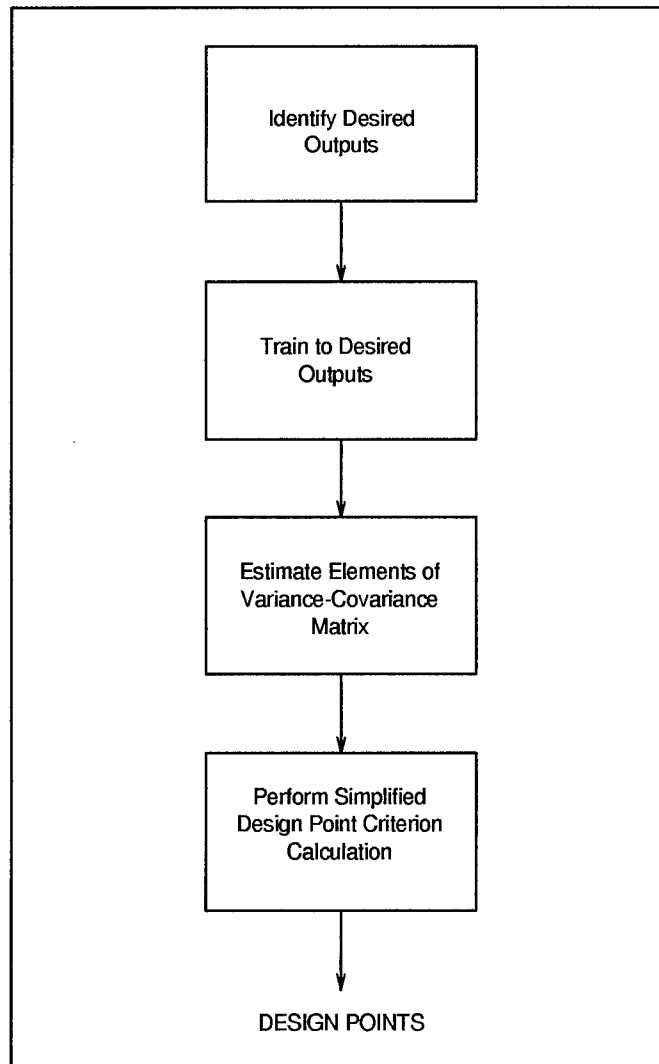
Figure 35. Simplified Multi-Response Design Point Determination Method
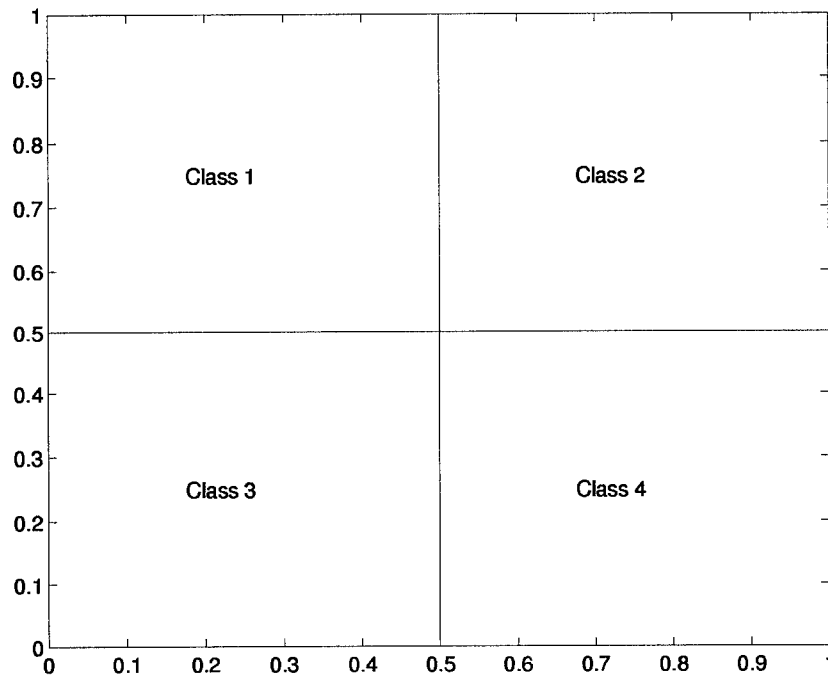
Figure 36. Four-Class Problem for Verification

The desired outputs for these exemplars were coded as in Table 5 with the first output deleted. The classification boundary produced by the multilayer perceptron was approximately equal to the true boundary ($\mathcal{E}_O = 0.0315$ for the training set). The variance-covariance matrix of the actual outputs is

$$
\begin{bmatrix}
3.1563 & -0.1565 & 0.1602 \\
-0.1565 & 8.1407 & -0.2212 \\
0.1602 & -0.2212 & 15.0875
\end{bmatrix}
\tag{159}
$$

which is approximately to a diagonal variance-covariance matrix. It appears that preselecting desired outputs is a viable method of arriving at actual outputs that are approximately uncorrelated.

In most discrimination problems, the multilayer perceptron will not be as well trained. In a more realistic problem, the four-class problem from Section 4.2 was used. The desired

116

outputs for only 50 input vectors were coded as in Table 5. A multilayer perceptron with ten hidden nodes was trained. The inverse variance-covariance matrix, $\hat{\Sigma}^{-1}$, estimated using the residuals from the trained network is

$$\hat{\Sigma}^{-1} = \begin{bmatrix} 14.3777 & -3.4741 & -8.2570 \\ -3.4741 & 123.4423 & 0.5783 \\ -8.2570 & 0.5783 & 12.5190 \end{bmatrix} \tag{160}$$

This matrix is not approximately diagonal. However, at least four of the elements could be considered 0 reducing the matrix to

$$\hat{\Sigma}^{-1} = \begin{bmatrix} 14.3777 & 0.0000 & -8.2570 \\ 0.0000 & 123.4423 & 0.0000 \\ -8.2570 & 0.0000 & 12.5190 \end{bmatrix} \tag{161}$$

This matrix will be different for every multilayer perceptron trained. Yet, one would expect approximately the same structure. The relatively large value of the $(2, 2)$ element stems from the class coding and the training. Notice that the second output is coded 1 for both Class 1 and Class 2 and 0 for Class 3 and Class 4. Since the multilayer perceptron can accurately discriminate between a super-class containing classes 1 and 2 and a super-class containing classes 3 and 4, the estimated variance of the second output is small. This small variance results in a large value in the inverse matrix.

The design criterion reduces to approximately

$$|D| = \left| 14.3777 F_{.1}^T F_{.1} - 8.2570 F_{.1}^T F_{.3} + 123.4423 F_{.2}^T F_{.2} - 8.2570 F_{.1}^T F_{.3} + 12.5190 F_{.3}^T F_{.3} \right|$$

$$\tag{162}$$

The original criterion with 16 terms has been reduced by 69 percent. The reduced criterion was used to determine 30 design points. Using the truth model, these 30 design points were
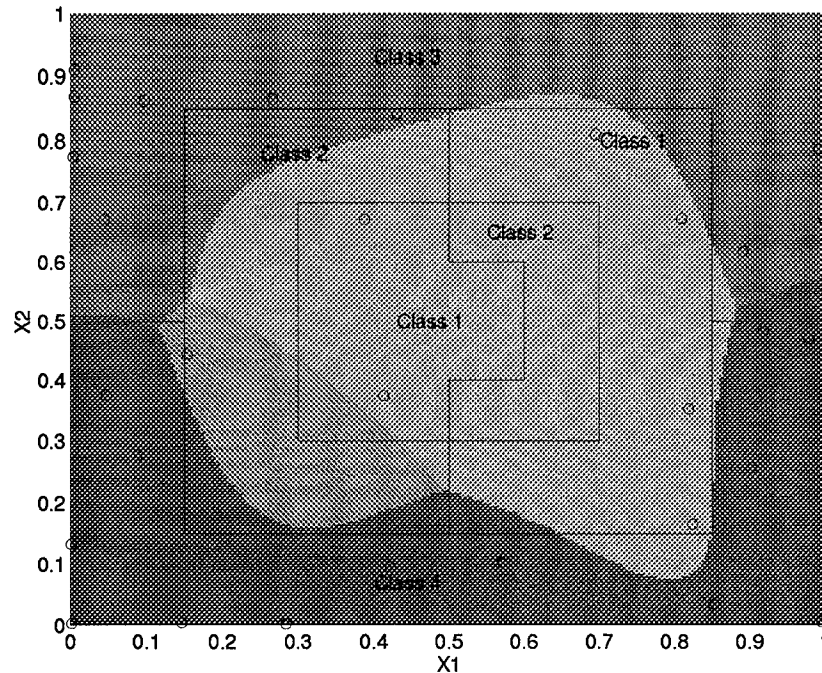
Figure 37. Multiple Output Discrimination Problem—Design Points and Resulting Boundary for Reduced Criterion

classified and added to the training set. Figure 37 shows the design points and the resulting multilayer perceptron boundary. Figure 38 compares the average test set classification error for the simplified criterion method, the "full" criterion method and a randomly selected set of points. Here, the error when using the simplified criterion method is significantly less than the error for the randomly selected points ($\alpha = .05$). Calculation of this simplified design point criterion required 12.87 system hours on a Sun Sparc station 5. This is a 31 percent reduction of the time required to process the original multiple output criterion.

4.4   Chapter Summary

In this chapter, sampling methods for multiple output multilayer perceptrons were developed. These statistically-based methods select design points for experimentation so as to

118
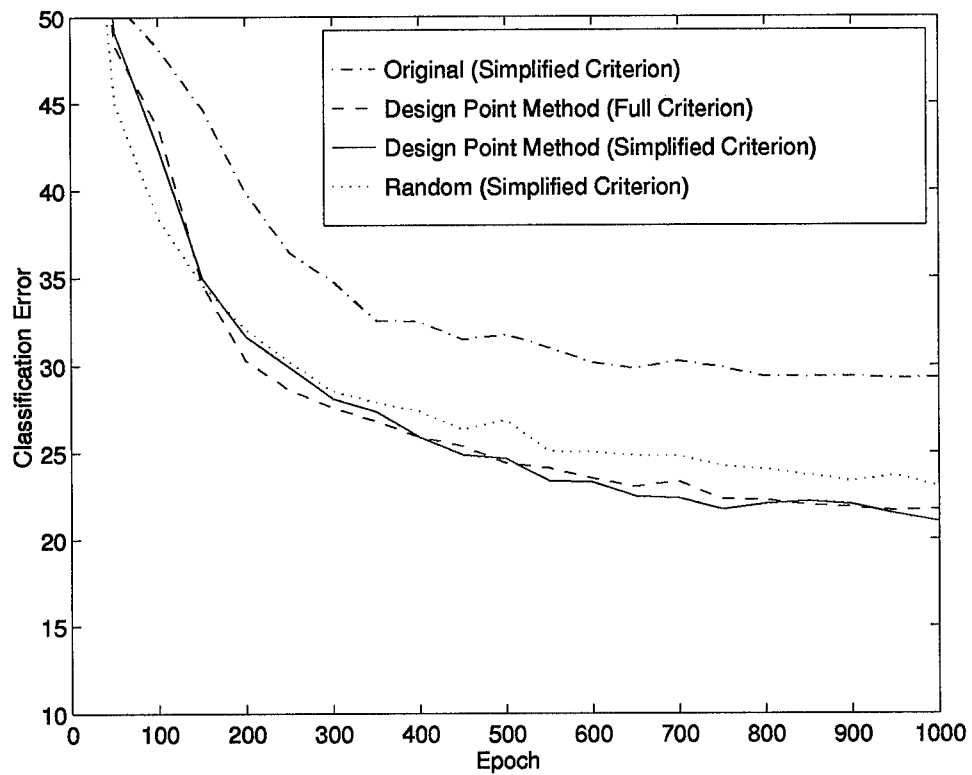
Figure 38. Average Classification Error Comparison for Reduced Multi-Response Criterion (30 Runs, Sampled Every 20 Epochs)

best estimate the multilayer perceptron weights. The methodologies presented in this chapter are combined into an overall approach and applied to a sample problem in Chapter V.

4.4.1   Multiple Output Criterion—Discrete Feature Space.   Initially, the multi-response criterion is investigated and found to simply be a weighted sum of individual single-response criteria. The weighting is done according to the inverse of the variance-covariance matrix.

In the single response case, an equation was derived for the D-optimal criterion given the addition of a single exemplar to a set of exemplars. Mitchell uses the result to define the discrete exchange algorithm. The corresponding multivariate derivation is not present in the literature. In this chapter, that extension was theoretically derived. Given the result, extension of the discrete algorithm to handle multi-class problems is relatively effortless.

4.4.2   Multiple Output Criterion—Continuous Feature Space.   The full design point criterion was tested for a four-class, two-dimensional problem. The D-optimal criterion resulted in a significantly lower average classification error over randomly chosen points. To show that the technique could be applied iteratively, the weights from the resulting multilayer perceptron were used to find a second set of exemplars. These points significantly reduced the average classification error over the first iteration and over a second set of random points.

In summary, a sampling method for multiple output multilayer perceptrons has been developed. It was shown empirically that more accurate multilayer perceptrons are produced when this method is used. In addition, the iterative nature of the method was illustrated.

4.4.3   Reducing the Complexity of Design Point Determination.   The original criterion is simple in form, however, it is computationally unwieldy. A method is developed which simplifies the criterion by causing the outputs of the multilayer perceptron to be uncorrelated. Then, since the variance-covariance matrix becomes diagonal, there are only r terms in the criterion as compared to r^2.

By judiciously choosing the values of the desired outputs, one can cause these outputs to be approximately uncorrelated. Using Hadamard matrices, it is shown that it is possible to determine the appropriate desired outputs for any number of classes. When tested, the accuracy of the resulting multilayer perceptron was comparable to the accuracy when using the full criterion. The conclusion is that the simplification is a viable method of significantly reducing the complexity of selecting exemplars.

V. Design of Experiments Methodology

5.1   Introduction

The purpose of this chapter is, first, to succinctly list the steps one would use to determine experimental settings for multilayer perceptrons classifiers and, second, to demonstrate the application of these steps with two examples.

5.2   Overall Methodology

5.2.1   Single Output (Univariate) Multilayer Perceptrons.   Figure 39 shows the overall design point methodology for single output multilayer perceptrons. This figure unites the research done in Chapter III into a single framework for practical implementation. The first consideration is whether the feature space will be considered discrete or continuous. It is possible for a classification problem with continuous data to be discretized and treated as a discrete feature space. This research has explored the Discrete Exchange Algorithm for determining design points in a discrete setting.

If the feature space is continuous, then a decision is made as to whether complexity reduction is desired. If the classification problem under study is relatively small with few parameters, it may be sufficient to use Powell's algorithm with no complexity reduction. However, if the problem is large with many weights, or if it is necessary to produce design points often, then some simplification should be considered.

If the user has some reason to believe that there are linearities in the classification problem, then it would be desirable to test a DLF network structure. By doing so, not only will the design point method be simplified, but the user may discover unknown linear characteristics of the problem. If the DLF network structure is indicated, then design points

122
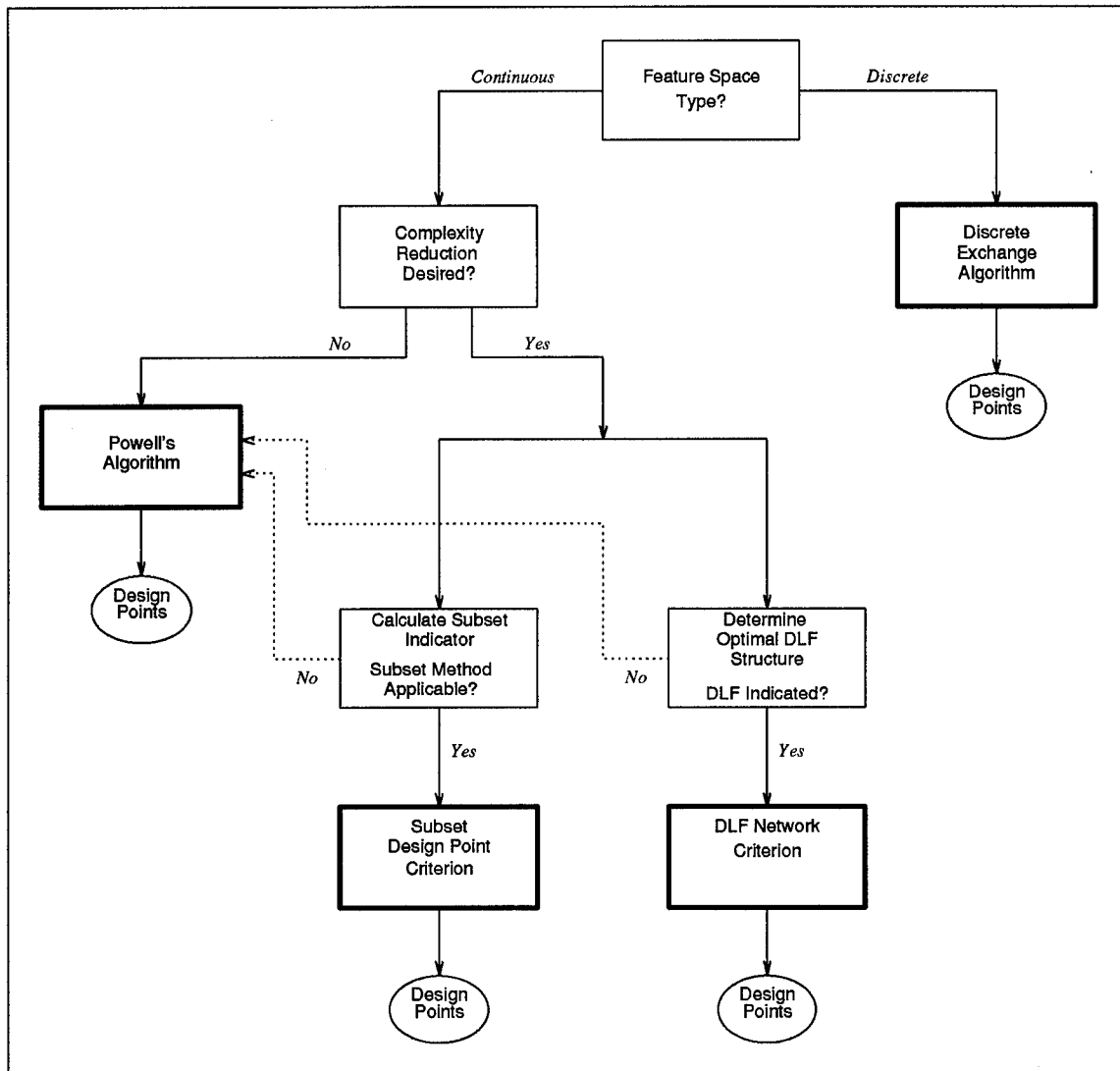
Figure 39. Overall Single Output Design Point Methodology

can be found using the DLF network criterion developed in Chapter III. If a DLF network is not indicated, then either the original criterion or a subset criterion must be used.

A quick calculation of the subset indicator developed in Chapter III will help a practitioner decide if the subset method is applicable. As stated earlier, it is believed that the subset method is applicable for all two-class problems.

There is no "correct" path through the flow diagram shown in Figure 39. The multilayer perceptron user must judge which method should be used from several factors. Some of these factors include:

- The size of the training and test sets.

- The complexity of the classification problem.

- The accuracy of the initial weight vector.

- The number of inputs and hidden nodes.

- The convergence of the maximization (or minimization) algorithm used to find design points.

- The frequency with which the design point method will be used.

5.2.2   Multiple Output (Multivariate) Multilayer Perceptrons. Figure 40 shows the overall multiple output design point methodology. This figure unites the research conducted on choosing design points for multi-output networks. As in the single output case, the first consideration is the type of feature space. If the feature space is considered discrete, then the Discrete Exchange algorithm with the multi-response criterion developed in Chapter IV should be used.

If the feature space is considered continuous, then the multi-response criterion is used within the Powell algorithm. If the problem is complex, a simplified criterion can be used
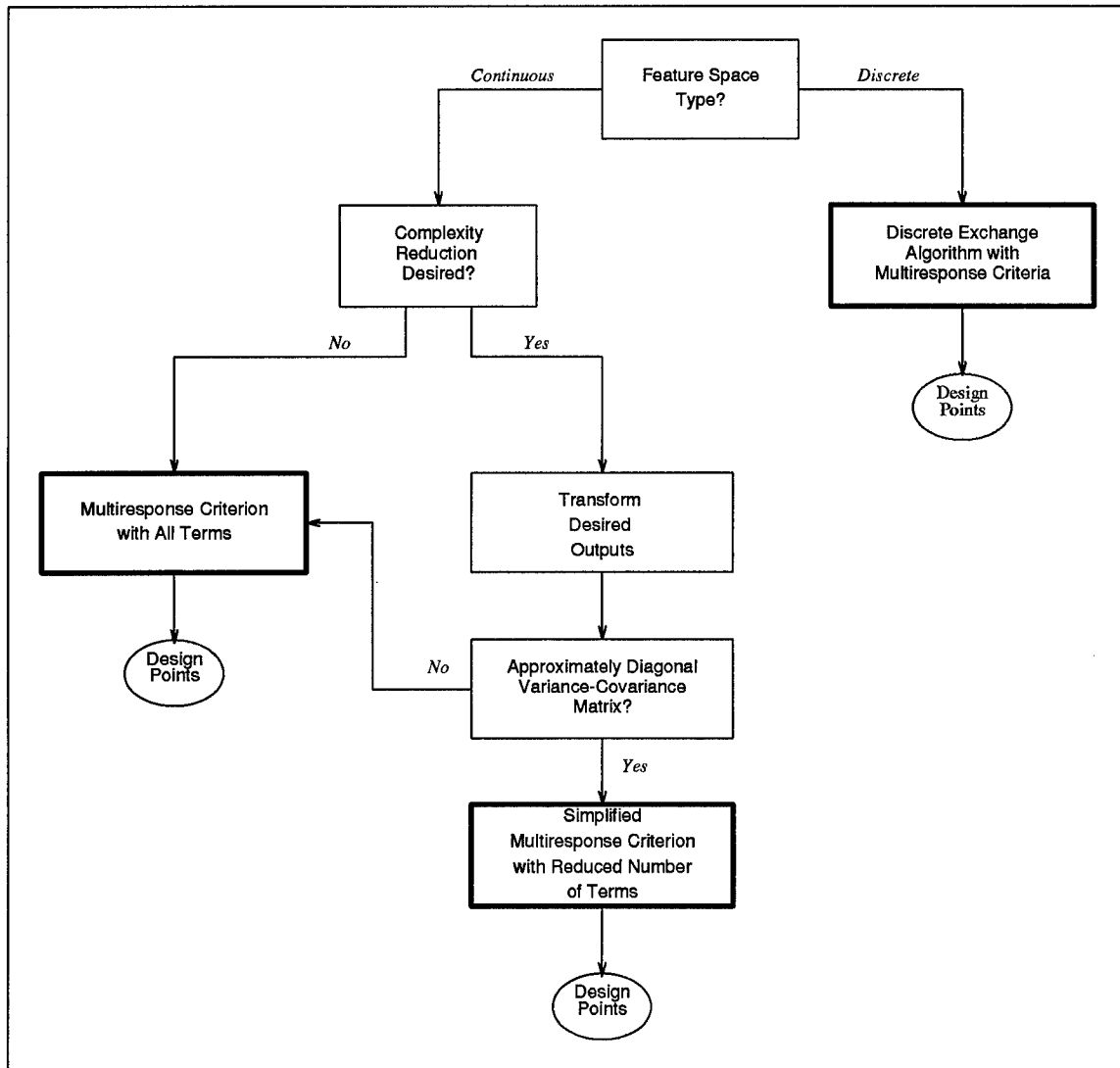
Figure 40. Overall Multiple Output Design Point Methodology

after a transformation of the desired outputs. This complexity reduction may be less desirable than the "full" method for the following reasons:

- For some problems, the transformation of the desired outputs may result in a less accurate multilayer perceptron (as compared to the more traditional coding).

- The interpretability of the network has been degraded. With a traditional coding scheme, an exemplar is coded as Class i if output node i has the largest value. In the transformed setting, one must look at all output values and compare them to desired outputs before the exemplar can be classified.

5.3 Application to Armor Piercing Incendiary Projectile Data

In this section, the overall methodology for single output multilayer perceptrons is applied to the area of aircraft survivability. It cannot be stressed enough that this is only one application where the methods outlined in this research can be used to determine experimental design points. *Whenever experiments are to be performed and a multilayer perceptron will be used to model the data, the results presented in this research are applicable.*

5.3.1 Background. An armor piercing incendiary projectile is a bullet that can perforate light armor. The projectile contains a flammable mixture that is generally encased in the nose of the projectile body. The design of the projectile allows the jacket over the nose to deform or peel off upon impact of the target skin. Consequently, the incendiary mixture will flash as the steel core of the bullet continues its flight [49].

The intense flash following jacket failure of an API is known as incendiary functioning (IF). At present, there are five classifications for incendiary functionings: complete, partial, slowburn, delayed and non-functioning. These classes are used extensively in isotropic material IF studies [37].

126

Prediction equations for API penetration mechanics are an essential part of the Air Force's aircraft vulnerability analysis program. Specifically, a projectile's type of incendiary functioning after aircraft skin penetration is used in aircraft survivability analysis. A problem currently facing the Air Force is that the prediction methodologies used for armor piercing incendiary projectiles do not extend to the use of composite materials.

The Survivability Enhancement Branch of Wright Laboratories (Wright-Patterson Air Force Base) performs test shots in order to gather the data to be used in the development of prediction models. To date, data has been collected using standard penetration mechanics testing sequences. These sequences are equivalent to factorial or half-factorial experimental designs. Recently, multilayer perceptrons have been used as models to predict the functioning of APIs based on the firing characteristics [37, 5, 6]. Given that multilayer perceptrons are to be used, the research presented in this document for designing experiments is extremely applicable.

Figure 41 shows the test used to obtain the experimental data. The feature set used for classification is organized into feature vectors, each vector representing the measurements and results for a single API projectile test shot. The features used were as follows:

1. **Striking Velocity** (V_s) – The projectile's velocity was measured immediately before impact and was assumed to be the striking velocity of the projectile on the target panel. Possible values: $\approx$1500 fps, $\approx$2000 fps, $\approx$2500 fps.

2. **Obliquity Angle** (OBL) – The obliquity angle is the angle between the line perpendicular to the panel surface and the projectile's flight path. For example, a shot fired straight at a panel has a zero degree angle of obliquity. In this analysis, OBL is converted to the secant of the angle (SECT). Possible values: 1.00000 (0 degrees), 1.15470 (30 degrees), 1.41421 (45 degrees), 2.00000 (60 degrees), 2.92380 (70 degrees).
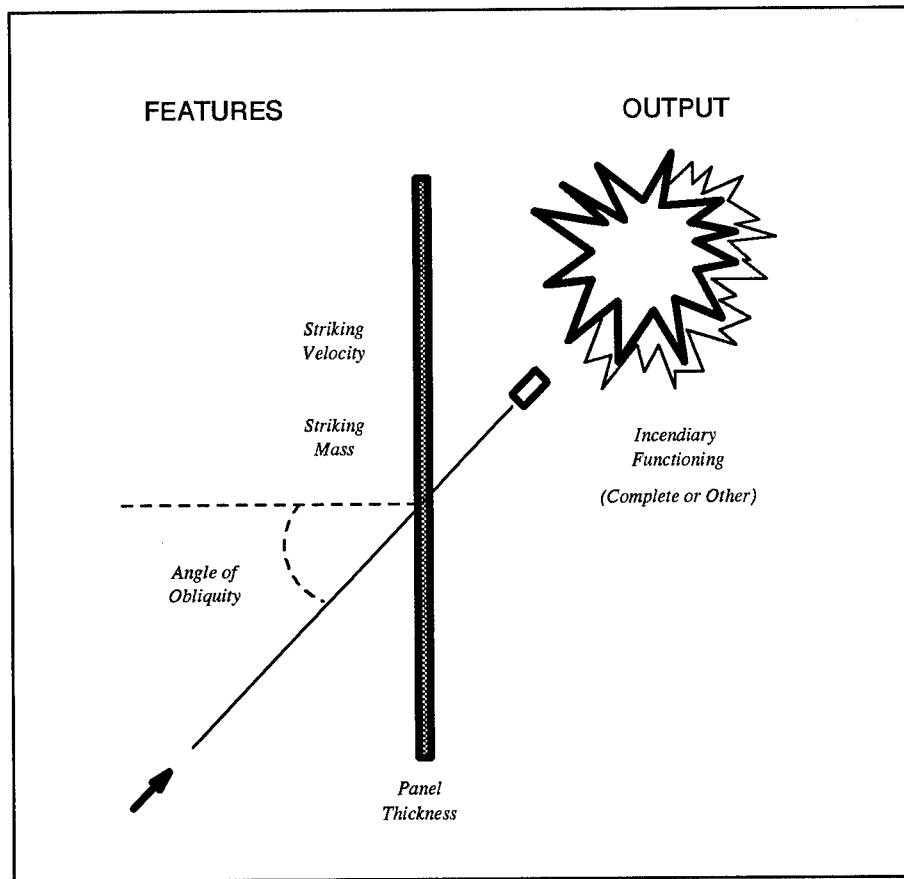
Figure 41. API Projectile Firing

3. **Striking Mass** (M_s) – The striking mass is assumed to be the mass of the projectile before firing and is measured in grains. Possible values: $\approx$745 grains, $\approx$945 grains.

4. **Panel Ply Thickness** ($TKIN$) – Test panels varied in thickness according to ply size. Possible values: 32 ply, 48 ply, 64 ply.

5. **Functioning** (IF) – For this analysis, projectile firings will be classified as "function" or "non-function."

5.3.2   *Application of Design Point Methodology.*

5.3.2.1   Discrete Feature Space. The design of experiments methodology introduced above was applied to the API projectile situation. It was assumed that the data from 50 API projectile experiments was available. A multilayer perceptron with 10 hidden nodes was trained using 30 training exemplars and 20 test exemplars. For the practical application of multilayer perceptrons, it is necessary to use a test set. As stated previously, the test set is used to test the accuracy of training while training is on-going.

One of the major advantages of allocating data to a test set is that it helps guard against overlearning. Classification error rates in Chapters III and IV were reported on the test set. In this chapter, since only 50 vectors are assumed to be available, an additional set—the validation set—will be used. The validation set should be viewed as a large set of control data not known during training, but used to judge the results of the method. The error rate on the validation set allows one to see how accurate the classifier would have been if more data was available.

A second practical consideration is normalization of the data. When initial weights are being obtained, the training and test sets are normalized. In this research, normalization is done by rescaling the data to values between 0 and 1. The elements of the sth un-normalized

129
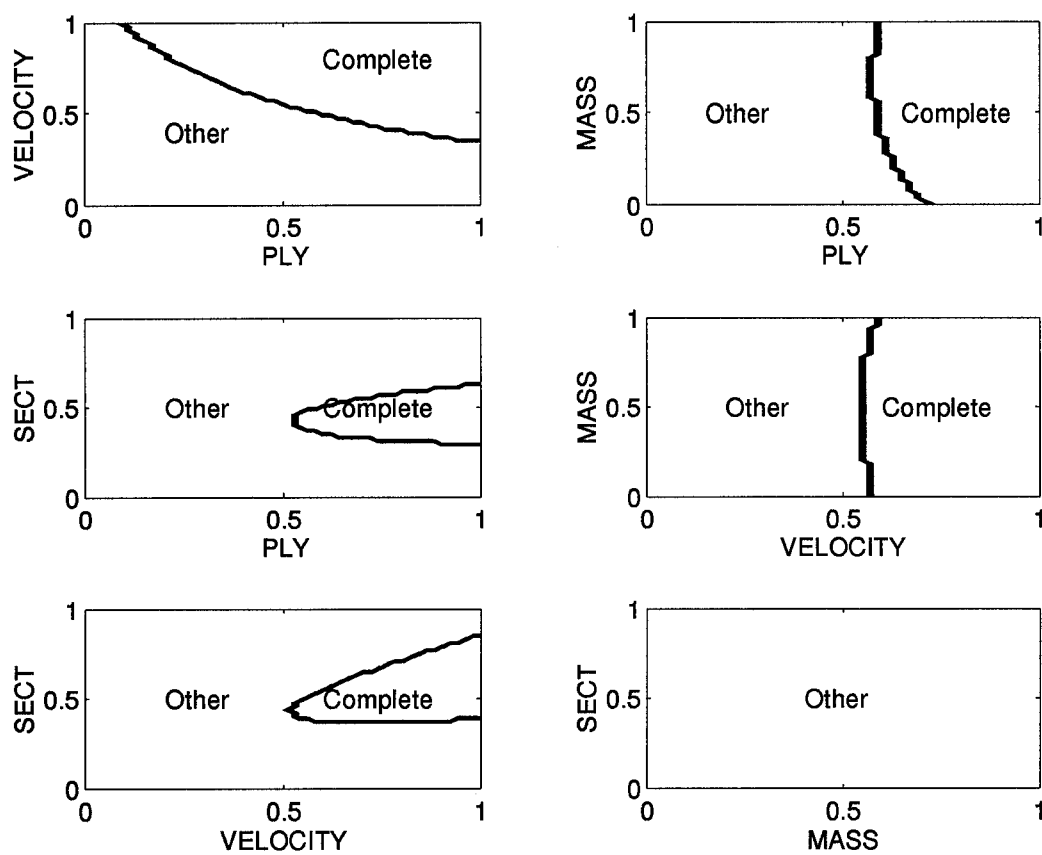
Figure 42. Original API Projectile Boundaries

feature vectors become:

$$\tilde{x}_k^s = \frac{x_k^s - \min_s x_k^s}{\max_s x_k^s - \min_s x_k^s} \qquad k = 1, \ldots, n; s = 1, \ldots, N \qquad (163)$$

where N is the total number of exemplars in the train and test sets, k denotes the feature, s denotes the exemplar and $\tilde{x}_k^s$ is the normalized element of the exemplar.

Since the initial set of weights is based on the normalized exemplars, the design point method will yield normalized design points. To perform experiments, these design points must be un-normalized. The values of $\min_s x_k^s$ and $\max_s x_k^s$ can be used to produce approximately un-normalized features.

Figure 42 shows the original multilayer perceptron boundaries. The weight vector resulting from this initial training (50 vectors) was used to choose 10 design points. Due to budgetary/resource constraints, it was not possible to actually conduct the indicated experiments. Instead, the 281 shots available in the WL/FIVS database were used to develop a multilayer perceptron to serve as the truth model and take the place of actual experiments. This truth model is shown in Figure 43. The 10 design points chosen were processed through this truth model and with the resulting classification were added to the training set (resulting in a total of 60 vectors). The design points and the resulting boundaries are shown for two features at a time in Figure 44.

The use of the test set proved to be especially advantageous in this application. Due to the small number of training vectors used, overlearning was observed during nearly every multilayer perceptron run. Therefore, simply averaging the classification error across multilayer perceptron runs does not reflect the true capability of the individual networks. To adjust, classification error rates were noted at the epoch where the test set classification error rate was at a minimum. The first plot in Figure 45 shows average classification errors at the minimum test set error for the train, test, and validation sets. The different shading on the bars denotes
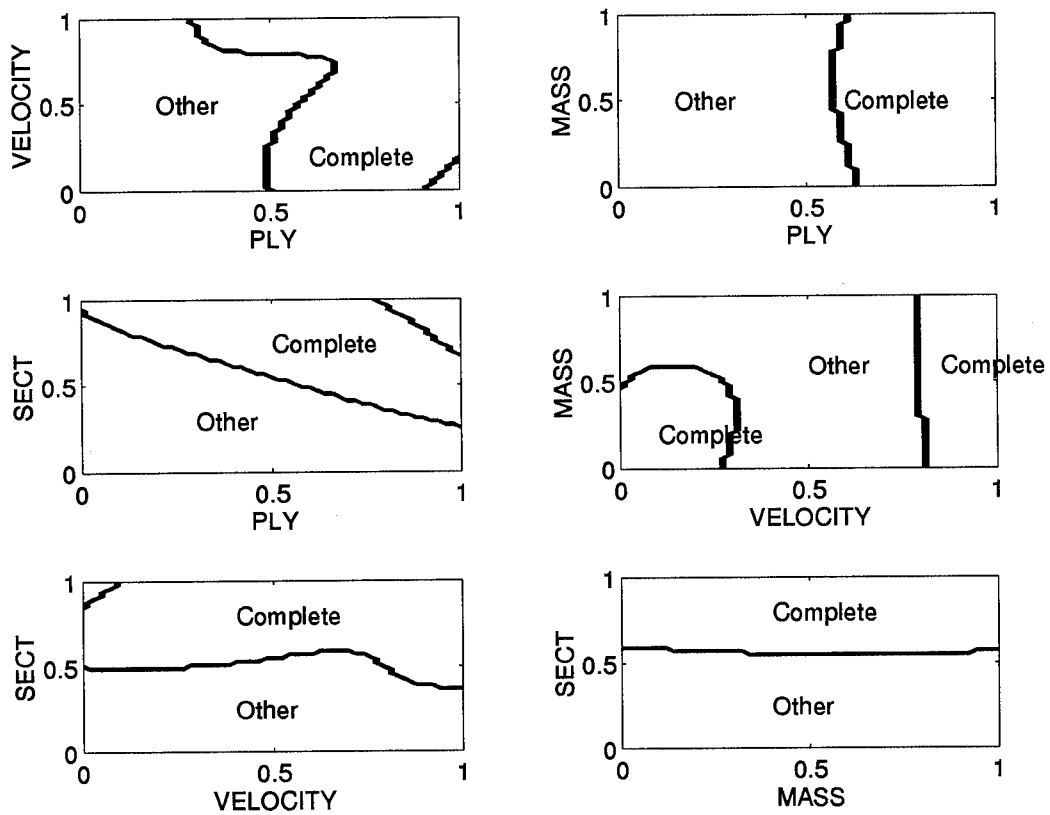
Figure 43. API Projectile Truth Model
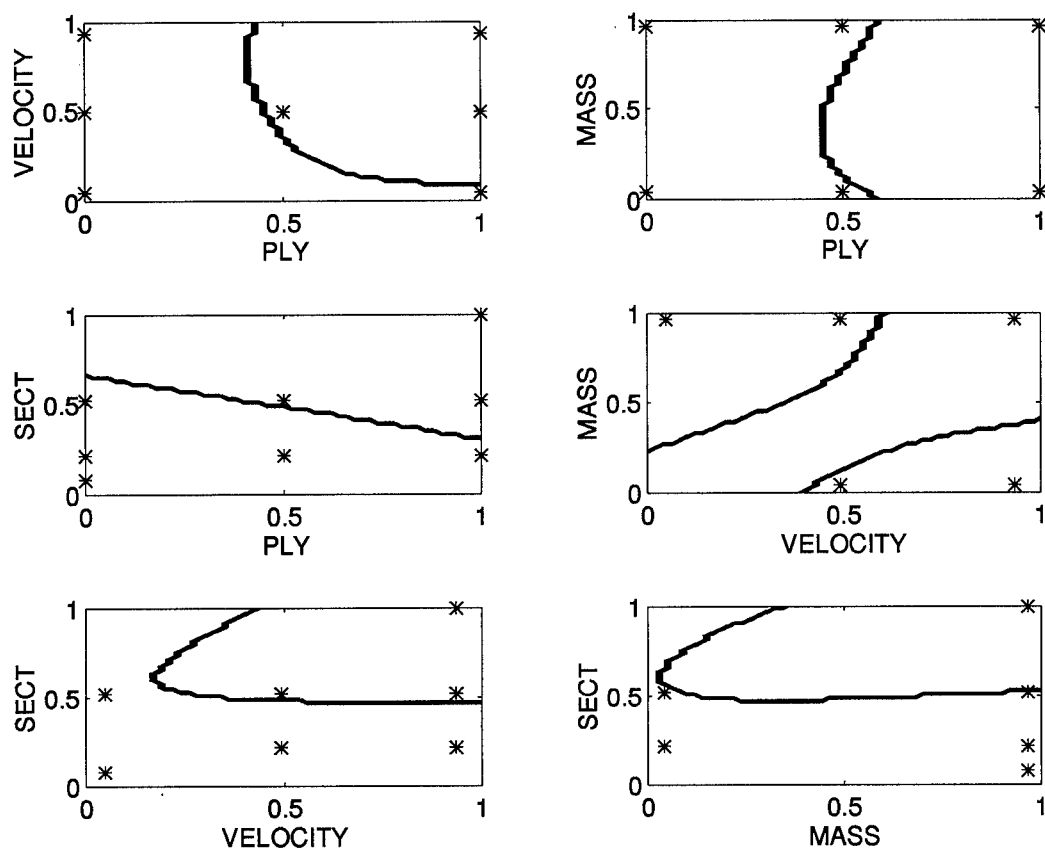
Figure 44. Design Points and Resulting API Projectile Boundaries—Discrete Feature Space
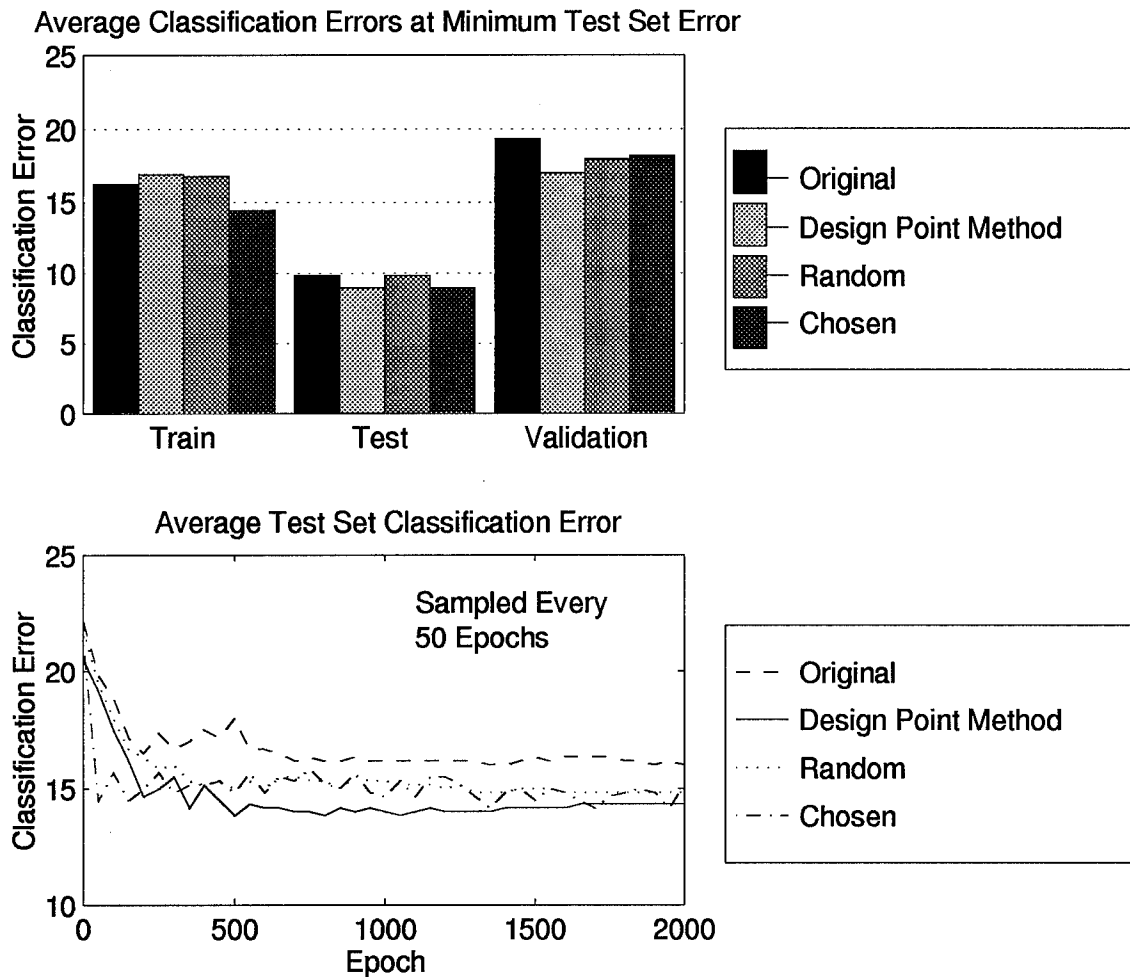
Figure 45.   API Projectile Average Classification Error Comparisons—Discrete Feature Space (30 Runs)

the average original error rate, the average error rate when the design point method was used, the average error rate when random points were used and the average error rate when points were chosen according to a user defined scheme.

In this case, the "chosen" design points were obtained by reducing the three-level feature PLY to a two level feature, the three-level feature VELOCITY to a two-level feature and the five-level feature SECT to a three-level feature. Saliency measures on preliminary multilayer

perceptron runs indicated that the feature MASS contributed little to classification, so this feature was fixed at its high value. These reductions result in the 12 vectors $(2 \cdot 2 \cdot 1 \cdot 3 = 12)$ that make up the "chosen" data set. This data set represents what is typically done at WL/FIVS when a limited number of experimental settings are required.

Note that over 30 runs the design point method yields a lower classification error on the validation set than the random points. The test set error is also lowest for the design point method. The difference in classification errors may appear small here. The reason is the small number of exemplars added to the training set (only 10). The next logical step would be to use the resulting weight vector and select another 10 exemplars.

The second plot in Figure 45 shows the average test set classification error over the 30 runs of the multilayer perceptron. (This plot compares to classification error plots in previous chapters.) The figure illustrates that, in general, the design point method will yield a lower error rate than the random points and the chosen points. However, it does not show the lowest achievable error rate because of the averaging.

5.3.2.2 Continuous Feature Space. The current experimental configuration for API shot equipment allows only for certain discrete values of the shot parameters. In this section, it is assumed that there is a need to treat the input features as continuous variables. Perhaps the test equipment was improved to provide a wide range of settings, or experimenters wish to know at what levels it would be profitable to conduct future tests.

All conditions are the same as in the previous section. First, the "full" criterion was used to find 10 design points in a continuous feature space. Next, the high value of the subset method indicator revealed that the method was applicable and 10 points were chosen using the subset criterion. These sets of 10 points were input to the truth model to obtain classifications. Multilayer perceptrons were then trained with the additional 10 points as part of the training set. Figure 46 shows the classification error results for the design point method, randomly

135
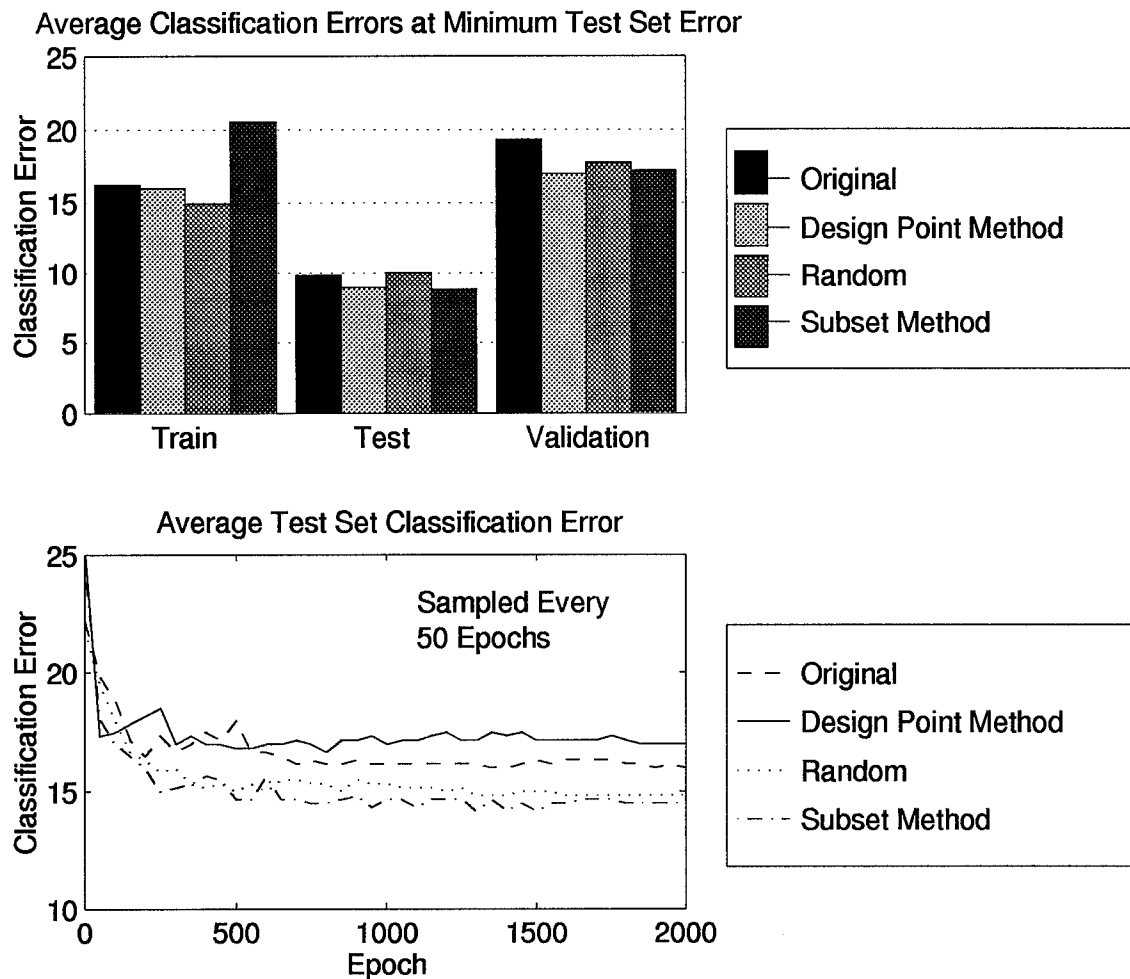
Figure 46.   API Projectile Average Classification Error Comparisons—Continuous Feature
            Space (30 Runs)

chosen points and the subset method. Note that the random points yielded the highest error
for both the test and validation sets (over 30 runs). The average test set classification error
(bottom of Figure 46) indicates that the subset method tends to yield the lowest classification
error over all 2000 epochs. This figure also indicates that the full design point criterion was
so sensitive to overlearning that it appears to produce a higher error rate than the original
multilayer perceptron.

Average Classification Errors at Minimum Test Set Error

Average Test Set Classification Error
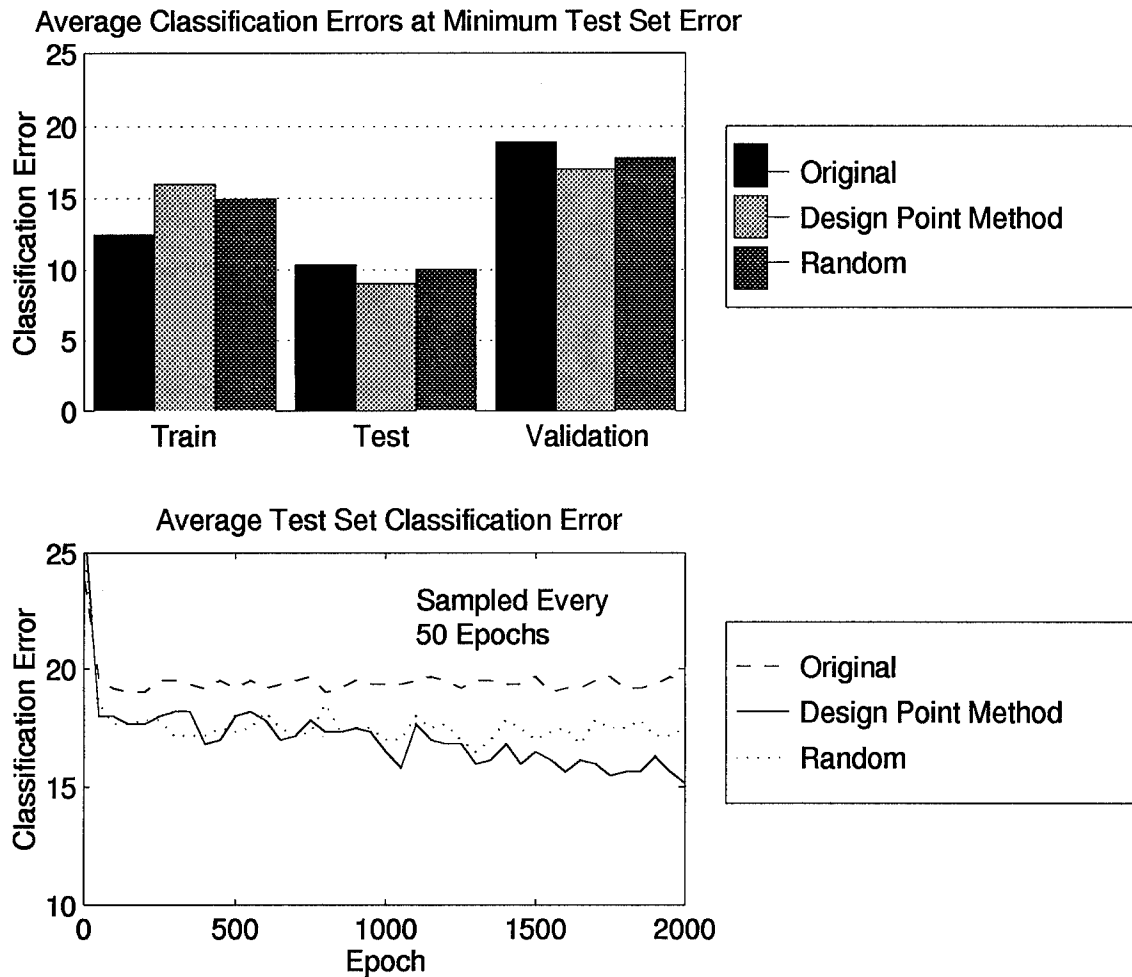
Figure 47.   API Projectile Average Classification Error Comparisons—Distributed Linear
Feedthrough (DLF) Network (30 Runs)

Finally, the DLF network with 6 middle nodes was applied to the same API problem.

Using the simplified criterion, 10 design points were chosen. As before, these design points

with their classifications were included in the training set. A DLF network, also with 6 middle

nodes, was trained on this data. In addition, a DLF network with 6 nodes was trained with 10

randomly chosen points added to the training set. Figure 47 shows the resulting classification

error rates. Again, the design point method yielded the lowest average classification error.

5.3.3   API Projectile Summary. The single output design of experiments methodology has been demonstrated for the API projectile classification problem. It was shown that the design point method developed in this research chooses points for experimentation which produce an accurate classifier. In the API projectile discrimination problem, 10 design points were added to an initial set of 50 vectors. If necessary, this method could be repeated to produce additional design points.

The research objective has been reached. This example has illustrated that an experimenter using multilayer perceptrons can now answer the question: Where do I perform the experiments?

5.4   Application to Stress-Time Plots for Predicting Incendiary Functioning Types

5.4.1   Background. In the last section, four parameters of an API projectile shot were used to predict whether functioning occurred. Although multilayer perceptrons provide an empirical model for predicting API projectile performance, several methods have been developed based on the physical characteristics of the system under study. In 1977, Falcon Research and Development (FRD) presented a model which has become the basis for the IF prediction methods used today [37].

FRD identified force (pressure opposing the projectile's penetration) and impulse (time force is applied) as the defining criteria for classifying IF types. FRD defined the stress (σ) that occurred for all normal obliquity shots. To account for stress at different obliquity angles, FRD developed $\triangle\sigma$'s for each type of projectile. The sum of the two stresses, σ and $\triangle\sigma$, equals the total stress effect. A time parameter t^* is used in place of impulse [23]. With $\sigma + \triangle\sigma$ and t^* calculated, stress-time plots are created. These plots delineate zones where the five IF classes would occur. Figure 48 provides an example of one of these plots.

138
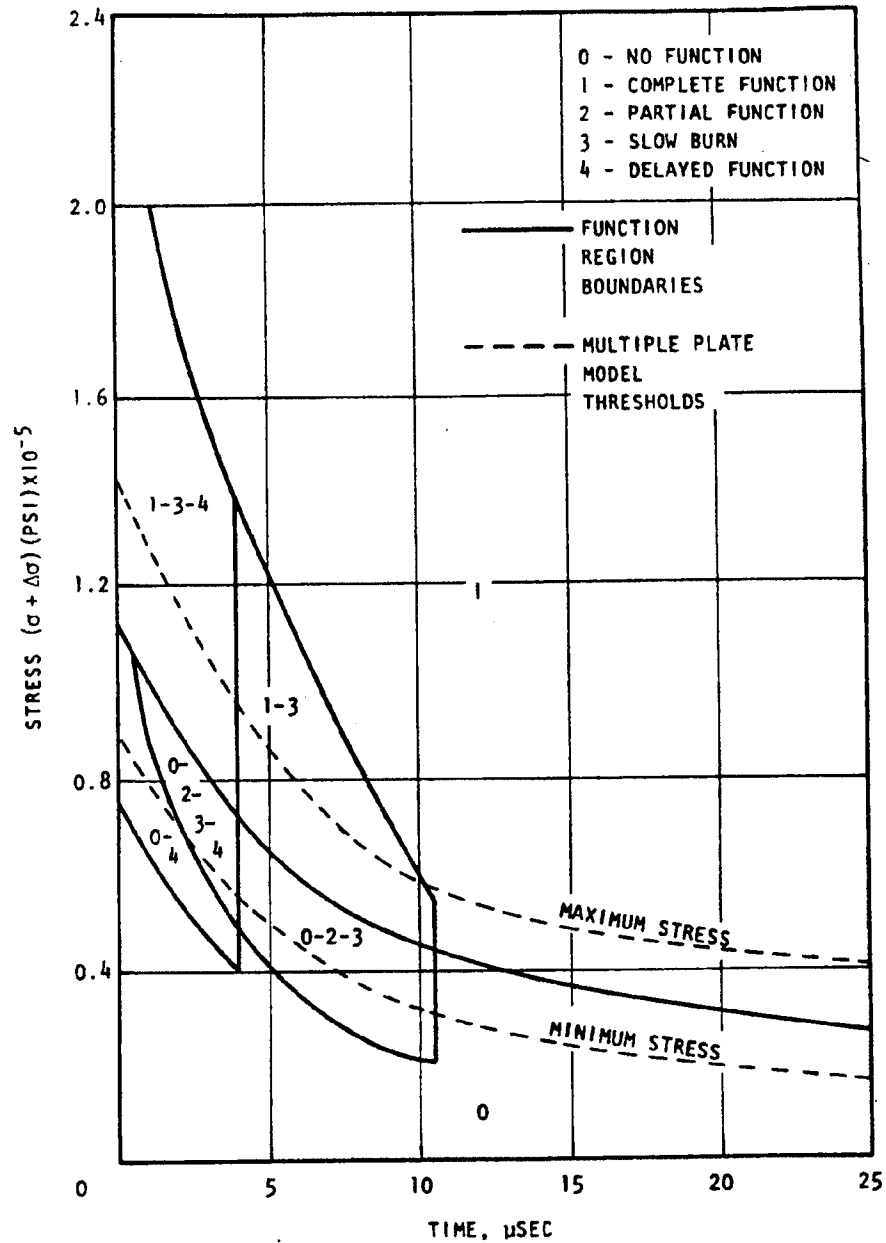
Figure 48. Falcon Research and Development Stress-Time Plot

Reprinted from [23]

139
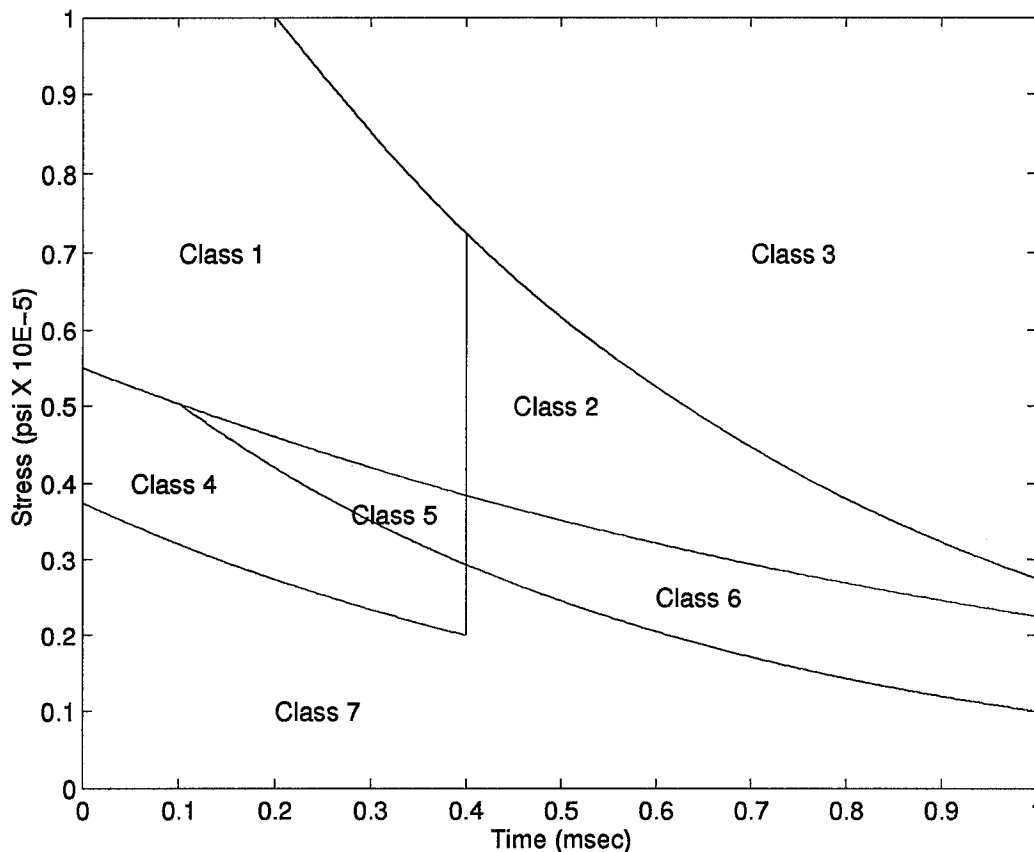
Figure 49. Truth Model for Seven-Class Stress-Time Plot Discrimination Problem

5.4.2 Application of Design Point Methods. The stress-time plot in Figure 48 was re-scaled and normalized to produce the plot shown in Figure 49. Each of the zones shown are numbered Class 1 through Class 7. This plot will be used as the truth model and the data will be generated from it.

To simplify the multi-response design criteria, the multilayer perceptron desired outputs were coded as shown in Table 10. Initially 100 vectors were used to train a multilayer perceptron with ten hidden nodes. Using residuals, the variance-covariance matrix of the outputs was calculated. Because the multilayer perceptron was not perfectly trained, the variance-covariance matrix was not approximately diagonal. However, it was possible to

140

Table 10. Desired Outputs for Seven Classes in Stress-Time Plot

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Output 1	1	1	1	1	1	1	1
Output 2	1	0	1	0	0.5	0.5	0.5
Output 3	1	1	0	0	0.5	0.5	0.5
Output 4	1	0	0	1	0.5	0.5	0.5
Output 5	1	1	1	1	0	0.8	0.8
Output 6	1	1	1	1	1	0	0.83333
Output 7	1	1	1	1	1	1	0

remove 18 of the 36 elements in the matrix. These elements were removed because they were relatively small and would have little effect in the design point criterion. The inverse variance-covariance matrix is shown below. The zeroes in this matrix indicate where elements were removed.

$$
\hat{\Sigma}^{-1} = \begin{bmatrix}
6.0117 & 0.0000 & -3.8187 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 12.5071 & 0.0000 & 0.0000 & 0.0000 & 4.6405 \\
-3.8187 & 0.0000 & 7.6334 & -3.1881 & 0.0000 & -3.5436 \\
0.0000 & 0.0000 & -3.1881 & 9.3183 & 0.0000 & 3.6686 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 10.7487 & 11.5373 \\
0.0000 & 4.6405 & -3.5436 & 3.6686 & 11.5373 & 82.8441
\end{bmatrix} \qquad (164)
$$

The resulting reduced design point criterion contains 63 percent fewer terms than the full criterion.

Using the reduced criterion, 30 design points were chosen and these points were added to the training set. Figure 50 shows the design points and resulting multilayer perceptron boundaries. Overlearning was not observed in this application as it was in the API projectile problem. Therefore, the average test set error rate was a good indicator of accuracy of the
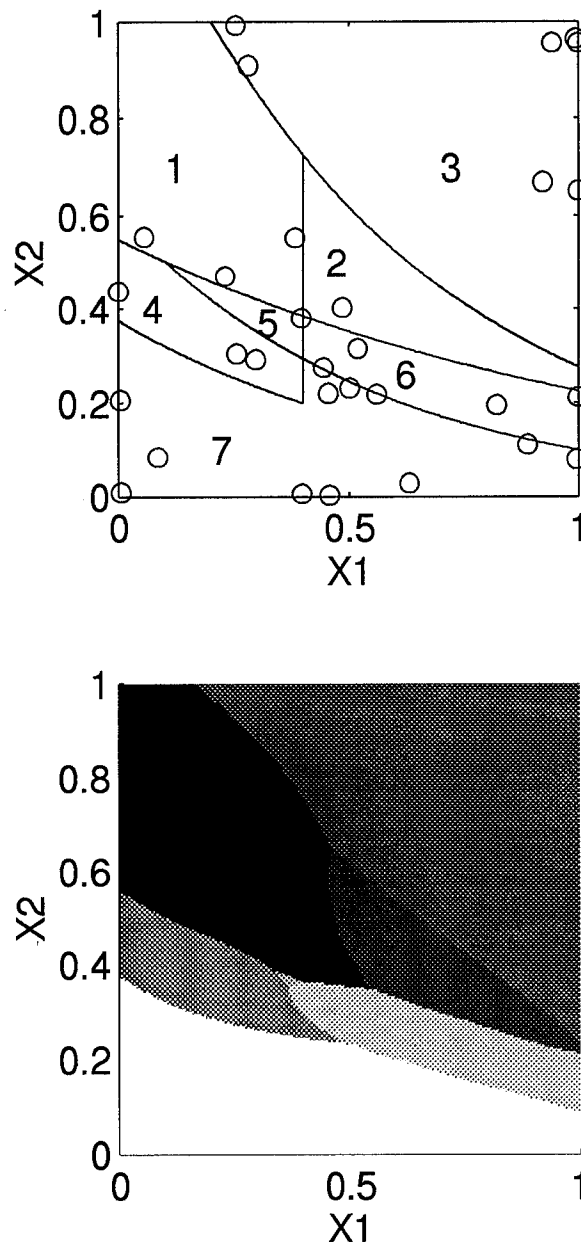
Figure 50.   Stress-Time Plot Problem—Design Points and Resulting Multilayer Perceptron
             Boundary

multilayer perceptron. The resulting average test set classification error is shown in Figure 51.

The design point method yields the lowest error rate.
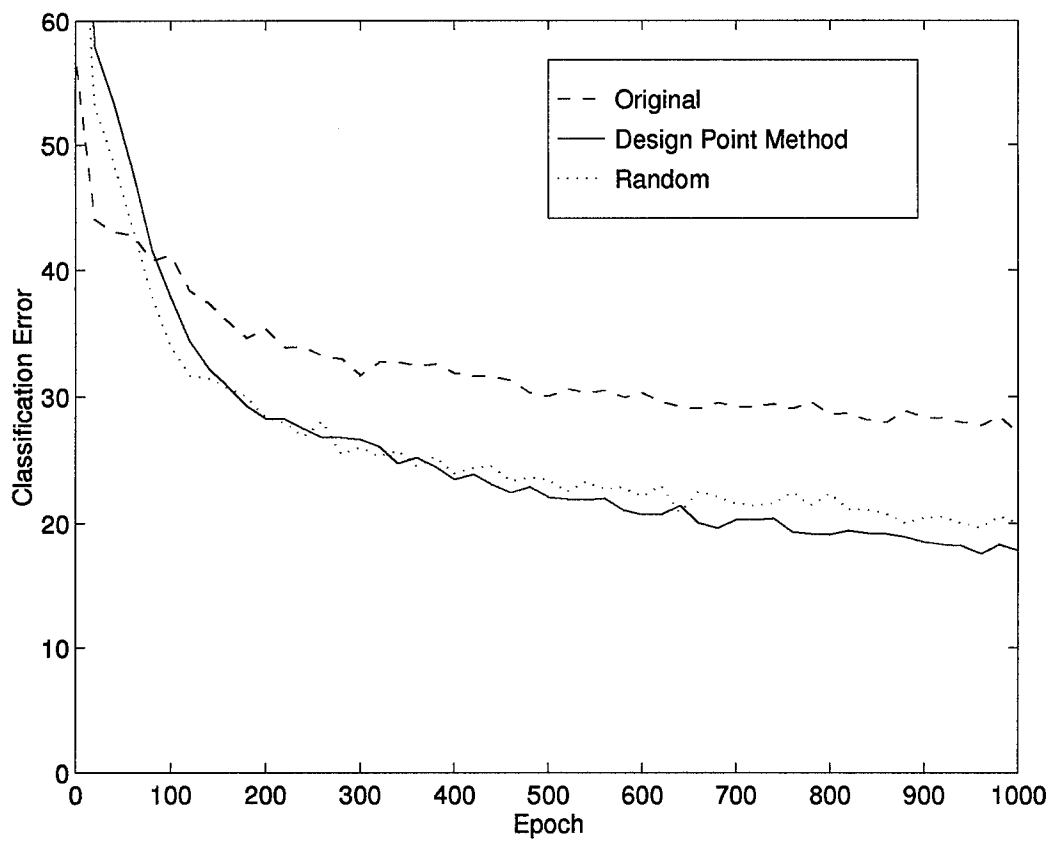
Figure 51. Stress-Time Plot Average Test Set Classification Error Comparisons (30 Runs, Sampled Every 20 Epochs)

5.5 Chapter Summary

Multilayer perceptrons are currently being used by WL/FIVS. Until this research was accomplished, no definitive technique for planning experimentation was available. In the past, as composite materials were procured, fractions of factorial designs were used. Selecting data this way does not consider the extensive nonlinearities in the multilayer perceptron structure and, as shown in this research, may not produce the most accurate classifier.

5.5.1 Application to Armor Piercing Incendiary Projectile Data.

The application of the single output methodology was successfully demonstrated for the classification of API projectiles. The full range of techniques was exercised and, in all cases, the newly developed techniques outperformed the other data selection methods.

Two practical issues applicable in "real world" problems were addressed. First, the importance of the test set was investigated for cases where small amounts of data are available. The inclusion of the test set helps guard against overlearning and insures near-optimal weights. Data normalization is the second practical issue. Normalized design points are selected by the design point method and, before experiments can be performed, the feature vectors must be restored from normalization.

5.5.2 Application to Stress-Time Plots.

The simplified multiple output methodology was successfully demonstrated on the seven-class stress-time plot discrimination problem. The desired outputs were transformed and the resulting actual outputs were approximately uncorrelated. The resulting points produced a classifier more accurate than random points.

VI. Summary and Recommendations

6.1 Summary

The significant contribution of this research is the introduction of new concepts and methods to the field of neural network training. Neural networks learn based on accumulated experience contained in a finite sample of cases with known outcomes (training exemplars). Construction of a neural network has two goals. First, the network must extract useful information from the training exemplars. The second goal is accurate classification of unseen data. The selection of the training exemplars is crucial to the creation of an accurate discriminator.

In this research, a statistically-based technique for selecting multilayer perceptron training exemplars is developed. The technique selects these training exemplars so as to best estimate the multilayer perceptron weights. At the core of the methodology is a criterion which estimates the size of the confidence ellipsoid for the weights in the multilayer perceptron. Minimization of this criterion over sets of training exemplars defines where future experimentation should take place. Until now, only graphical and heuristic algorithms were available.

The minimization of the criterion requires different methods depending on the whether the feature space is continuous or discrete. Powell's algorithm, a first order optimization method, is used for the continuous feature space and a discrete exchange algorithm is used for the discrete feature space.

6.1.1 Single Output Multilayer Perceptrons.
The single output method is shown empirically to produce more accurate classifiers than other data selection techniques. Specifically, multilayer perceptrons trained with exemplars determined by the design point criterion are more accurate than those trained with randomly selected exemplars. Additionally, exemplars

145

chosen in a "grid" or pattern produce less accurate classifiers than the exemplars chosen with the design points criterion. This "grid" pattern is often called a factorial design and is used quite extensively for linear models. The results here indicate that applying linear designs to circumstances where multilayer perceptrons are to be used may not be an effective strategy.

Once a set of design points has been chosen, two methods of ranking design points are presented. One method ranks the points according to a simple dot product and the other uses a saliency measure. The dot product measure can be interpreted as the squared length of the gradient of the output with respect to the network weights evaluated at the design point under consideration. This measure can be shown to be equivalent to choosing the point at which the variance of the predicted response is at a maximum. The saliency measure was derived from the original saliency metric proposed by Ruck [57]. The new measure ranks exemplars according to the total change in the output due to changes in the inputs. Considered together, these measures provided a comprehensive picture of the importance of each of the chosen exemplars. The ranking methods were shown to result in the selection of the "best" design points.

A large network can render a neural network method unusable. In this research, the complexity issue is successfully dealt with by introducing two simplifications which may be used if indicated. First, the distributed linear feedthrough (DLF) network adds linear connections to the traditional multilayer perceptron. The design point criterion is independent of the weights on these linear connections and, therefore, the criterion can be simplified. The DLF network has been applied previously. However, using the structure for reducing the complexity of determining design points is a new approach. A systematic procedure for assessing the amount of simplification possible is also introduced.

The second method for simplifying the determination of the design points emphasizes only the lower layer weights, once again decreasing the computational complexity of the

criterion. The use of only a subset of the model parameters for selecting design points has been tried for very simple nonlinear models and the application to neural networks is an original approach. An indicator is derived which allows a user to judge if the method is appropriate. This indicator estimates the correlation between a criterion considering the lower layer weights and a criterion considering all the weights. In Chapter III, an example is used to illustrate the use of both the DLF network and the subset method and in each case, more accurate classifiers are developed as compared to randomly chosen exemplars.

The final topic addressed for single output multilayer perceptrons is the effect of the initial weight vector on the method. It is shown that, as the number of initial exemplars increases (in other words, the more accurate the initial weight vector is), the benefit gained from the added design points decreases. An example was given in which the entire feature space was erroneously identified as one class. Design points were placed in a seemingly random pattern over the feature space, demonstrating the robustness of the method.

6.1.2 Multiple Output Multilayer Perceptrons. Extending the single output method to address any number of classes, as expected, increases the complexity of the technique. The criterion is now weighted by the variance-covariance matrix of the outputs of the multilayer perceptron. If r is the number of outputs, then the multiple output criterion has r^2 more terms. A four-class example problem illustrates the technique with all the terms of the criterion included. The exemplars chosen produce a more accurate classifier as compared to randomly selected points. To show that the technique can be applied iteratively, a second set of exemplars is chosen which further increases the accuracy of the multilayer perceptron.

Previously, the discrete exchange algorithm was defined only for the univariate (single output) case [44]. Researchers in this area had not performed the required multivariate extension (namely calculating the effects on the criterion of adding a single multivariate exemplar to a given set of exemplars). In this research, that extension is theoretically derived.

147

Given the result, extension of the discrete exchange algorithm to handle multi-class problems is relatively effortless.

Since the multiple output criterion depends on the elements of the variance-covariance of the outputs, reducing any of the elements in this matrix to zero reduces the number of terms in the criterion. By judiciously choosing the values of the desired outputs, one can cause the actual outputs to be approximately uncorrelated. (See Section 4.3.1.) Transforming the desired outputs requires a small change in how the outputs are interpreted. However, for the cases tested there was no degradation in performance. With the outputs uncorrelated, the variance-covariance matrix is diagonal and there are r terms in the criterion as opposed to r^2. The approach is simple and can be implemented for discrimination problems with any number of classes. This simplification is applied to a four-class problem with results very similar to results obtained with the full criterion.

6.1.3 Application of Methods To demonstrate the possible uses for the techniques developed, two application problems were used. The first discrimination problem predicts the performance of an armor piercing incendiary projectile striking composite materials. Four characteristics of the firing are used—thickness of the target, velocity of the projectile, mass of the projectile and the angle that the projectile strikes the target. Also discussed in this application are data and normalization considerations which must be dealt with in "real world" implementations. The full range of techniques developed for single output problems was exercised. In all cases, the newly developed techniques outperformed traditional methods of data selection.

The second discrimination problem exercises the procedures developed for multiple output problems. Stress-time plots delineate seven regions where different levels of incendiary functioning occur. The simplified multiple output method was used to select design points for future experimentation resulting in an accurate classifier.

148

Multilayer perceptrons are currently being used by WL/FIVS. Until this research was accomplished, no definitive technique was available for planning experimentation as more composite materials became available.

6.2 Recommendations

There are related research topics which could not be adequately covered within the scope of this research effort. Research in these areas would benefit design of experiments methodologies specifically and the training of multilayer perceptrons in general.

6.2.1 Ranking Methods for Training Sets.

The first research topic is an extension of the ranking measures developed in Chapters III and IV. Rather than ranking only design points, it may be possible to rank order the exemplars for any subsequent training set and only train on the highly ranked vectors. A procedure such as this would be especially useful if one believes there are "useless" or "unimportant" vectors in the training set. Also, in cases where training time is limited, ranking exemplars would allow for training only on those vectors containing the most information.

6.2.2 New Ranking Measure.

The second research topic also concerns the ranking measures developed in Chapters III and IV. The first measure ($\mathcal{M}_1$) ranks exemplars based on changes in the output with respect to changes in the weights, while the second measure ($\mathcal{M}_2$) ranks exemplars based on the changes in the output with respect to changes in the input exemplars. Some combination of these measures is required. This combination would identify exemplars that are "good" both in the weight-sense and in the feature-sense. Perhaps some mathematical connection between these measures can be defined, or the relationship may have to be heuristic.

6.2.3 Subset Criterion. Further investigation of the design of experiments emphasizing a subset of the multilayer perceptron weights is required. It may be that only the lower layer weights are necessary for all single output problems. In addition, the subset method should be extended to the multiple output case. The benefit to be gained from this research is a major simplification of the design point criterion.

6.2.4 Potential Extensions to Other Application Areas. This research barely scratched the surface of the potential application areas for the methodologies developed. Whenever a neural network user has control over the collection of training data, the data can be selected using these methods. Other areas where the procedure can be used need to be explored. It may also be possible to use the methods to reduce large sets of training data to smaller, more manageable, yet "information-rich" training sets. These potential uses should be investigated.

Appendix A. *Experimental Design—Theorems and Definitions*

A.1 *Proportionality of Volume of Confidence Region and $|F.^T F.|$*

Theorem 4 *Let the measured response of the ith experiment be given by*

$$y_i = \eta_i + e_i \tag{165}$$

where η_i is the true response and the error e_i is normally distributed. Then, the volume of the confidence region in the parameter space is inversely proportional to the value of the determinant $|F.^T F.|$.

Proof: The boundary of an asymptotic confidence region for θ with confidence coefficient $1 - \alpha$ is formed by [45]

$$\{\theta : (\theta - \hat{\theta})^T F.^T F.(\theta - \hat{\theta}) = ps^2 F_\alpha(p, \nu)\} \tag{166}$$

Let $\mathbf{A} = F.^T F.$ which is positive definite and $\delta > 0$. Then

$$\{\theta : (\theta - \hat{\theta})^T \mathbf{A}(\theta - \hat{\theta}) = \delta^2\} \tag{167}$$

represents an ellipsoid. This ellipsoid can be rotated so that its axes are parallel to the coordinate axes. The rotation is accomplished by an orthogonal matrix $\mathbf{T}$ where $\mathbf{T}^T \mathbf{AT} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$ and the λ_i are the eigenvalues of $\mathbf{A}$.

Let $(\theta - \hat{\theta}) = \mathbf{Ty}$. Then,

$$\mathbf{y}^T \mathbf{T}^T \mathbf{ATy} = \lambda_1 y_1^2 + \lambda_2 y_2^2 + \dots + \lambda_m y_m^2 = \delta^2 \tag{168}$$

Setting all the y_i's equal to zero except y_r yields $a_r = \frac{\delta}{\sqrt{\lambda_r}}$ as the length of the rth semi-major axis, $r = 1, \dots m$.

The volume of the ellipsoid is then given by [59]

$$v = \frac{\pi^{\frac{m}{2}}}{\Gamma(\frac{m}{2}+1)} a_1 a_2 \ldots a_m \tag{169}$$

$$= \frac{\pi^{\frac{m}{2}} \delta^m}{\Gamma(\frac{m}{2}+1)(\prod_i \lambda_i)^{\frac{1}{2}}} \tag{170}$$

$$= \frac{\pi^{\frac{m}{2}} \delta^m}{\Gamma(\frac{m}{2}+1)|\mathbf{A}|^{\frac{1}{2}}} \tag{171}$$

$$\propto |\mathbf{A}|^{-\frac{1}{2}} \tag{172}$$

So, minimizing the volume of the asymptotic confidence interval is equivalent to maximizing $|\mathbf{A}| = |F.^T F.|$ [59].

A.2 Correspondence of Generalized Inverse and $|F.^T F.|$

Definition 1 *The generalized variance of p variables with variance-covariance matrix $\mathbf{C}_p$ is given by $|\mathbf{C}_p|$ [62].*

The estimate of the asymptotic variance-covariance matrix of $\hat{\theta}$ is given by $(F.^T F.)^{-1} \sigma^2$. Therefore, the generalized variance of the parameters is given by

$$|\mathbf{C}_p| = |(F.^T F.)^{-1} \sigma^2| \tag{173}$$

$$= \sigma^2 |(F.^T F.)^{-1}| \tag{174}$$

where σ^2 is constant. It is clear that minimizing the generalized variance is equivalent to minimizing $|(F.^T F.)^{-1}|$, the suggested criterion.

A.3 Correspondence of $|F.|$ and Volume of Simplex in $F.$-Space

First, it will be shown that the 2×2 determinant is the area of a certain parallelogram.

Definition 2 *The quadrilateral with vertices* $O = (0,0), U = (u_1, u_2), V = (v_1, v_2)$ *and* $W = (w_1, w_2)$ *is a parallelogram if and only if*

$$w_1 = u_1 + v_1 \quad \text{and} \quad w_2 = u_2 + v_2, \quad \text{that is} \quad W = U + V \tag{175}$$

Use the symbol $P(U,V)$ *to denote the parallelogram with vertices* $O, U, V,$ *and* $U + V$. *Let* $M(U,V)$ *denote the matrix with* U *and* V *as its rows:*

$$M(U,V) = \begin{bmatrix} u_1 & u_2 \\ v_1 & v_2 \end{bmatrix} \tag{176}$$

Lemma 1 *If* t *is any real number, then* $\text{Area} P(U,V) = \text{Area} P(U, V + tU)$ *[60].*

Proof: This lemma can be proved by the following facts

1. $P(U, V + tU)$ is the parallelogram with vertices $O, U, V + tU$ and $U + (V + tU)$.

2. For any real number t, the point $V + tU$ lies on the line L through V and $U + V$. This is so because $tU = (tu_1, tU_2)$ lies on the line M through O and U, and hence, the fourth vertex must be $V + tU$ from the definition of a parallelogram.

3. The two parallelograms $P(U, V)$ and $P(U, V + tU)$ share the same base and have the same height so their areas are the same.

Theorem 5 $\text{Area} P(U,V) = \pm \det M(U,V)$

Proof: The reason that this result is true is that the row operations used to calculate the determinant do not change the area. More precisely, suppose $u_1 \neq 0$. Choose t such that $V^T = V + tU$ lies on the vertical axis, i.e., $v_1 + tu_1 = 0$. (See Figure 52.) Then from the lemma

$$\text{Area} P(U,V) = \text{Area} P(U, V + tU)$$

$$\det M(U,V) = \det M(U, V + tU) \tag{177}$$

153
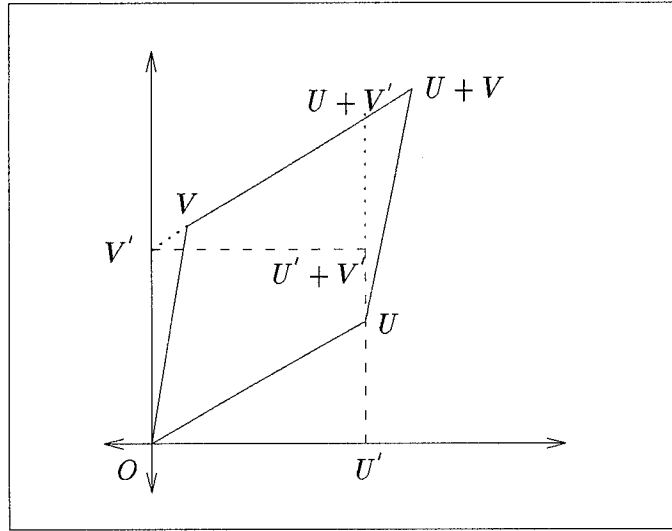
Figure 52. Translation of Parallelogram to Rectangle

Next, choose s so that $U^T = U + sV^T$ lies on the horizontal axis. Then

$$\text{Area}P(U, V^T) = \text{Area}P(U + sV^T, V^T)$$

$$\det M(U, V^T) = det M(U + sV^T, V^T) \tag{178}$$

The new parallelogram $P(U^T, V^T)$ is a rectangle with $U^T = (a, 0)$ and $V^T = (0, b)$ and we have

$$\text{Area}P(U^T, V^T) = ab = \det M(U^T, V^T) \tag{179}$$

Neither the area nor the determinant have changed, so:

$$
\begin{aligned}
\text{Area}P(U,V) &= \text{Area}P(U^T, V^T) \\
&= \det M(U^T, V^T) \\
&= \det M(U,V) \tag{180}
\end{aligned}
$$

If $u_1 = 0$ and $v_1 \neq 0$ we can interchange U and V. A similar result holds when the angle from U to V is negative.

In a similar manner, it can be shown that for any $n \times n$ matrix that the determinant is the volume of a certain parallelepiped. The volume of the simplex formed by the rows of the $n \times n$ matrix is proportional to the volume of this parallelepiped. This leads to the following theorem:

Theorem 6 *The value of $|F.|$ is proportional to the simplex formed by the origin and the p row vectors (experimental design points) in the $F.$-space.*

Particular attention must be paid to the fact that the elements of $F.$ are the partial derivatives of the given nonlinear response function with respect to the individual parameters— not the design point space. Therefore, this result holds in the $F.$-space only.

Appendix B. *Maximization of $|F^T F|$ with the Augmentation of an Additional Design Point*

Theorem 7 *Let A be a $p \times p$ matrix and u a $p \times 1$ column vector, then*

$$(A + uu^T)^{-1} = A^{-1} - \frac{A^{-1}uu^T A^{-1}}{1 + u^T A^{-1}u} \tag{181}$$

Proof: The outline of the following proof is given by Dykstra [22]. First, multiply the equation on the right by $(A + uu^T)$ and simplify as follows

$$(A + uu^T)^{-1}(A + uu^T) = \left(A^{-1} - \frac{A^{-1}uu^T A^{-1}}{1 + u^T A^{-1}u}\right)(A + uu^T)$$

$$I = A^{-1}A + A^{-1}uu^T - \frac{A^{-1}uu^T A^{-1}A}{1 + u^T A^{-1}u} - \frac{A^{-1}uu^T A^{-1}uu^T}{1 + u^T A^{-1}u} \tag{182}$$

Then, since $u^T A^{-1}u$ is a scalar,

$$\begin{aligned}
I &= I + A^{-1}uu^T\left[I - \frac{I}{1 + u^T A^{-1}u} - \frac{(u^T A^{-1}u)I}{1 + u^T A^{-1}u}\right] \\
&= I + A^{-1}uu^T\left[\frac{I + (u^T A^{-1}u)I - I - (u^T A^{-1}u)I}{1 + u^T A^{-1}u}\right] \\
&= I - 0 \tag{183}
\end{aligned}$$

Theorem 8 *Let A be a $p \times p$ matrix and u a $p \times 1$ column vector, then*

$$|A + uu^T| = |A|(1 + u^T A^{-1}u) \tag{184}$$

Proof: This proof is also outlined by Dykstra [22]. Multiply the result given in the theorem above on the right by A using $(A + uu^T - uu^T)$ for the left hand side:

$$(A + uu^T)^{-1}(A + uu^T - uu^T) = \left(A^{-1} - \frac{A^{-1}uu^T A^{-1}}{1 + u^T A^{-1}u}\right)A$$

$$I - (A + uu^T)^{-1}uu^T = A^{-1}A - \frac{A^{-1}uu^T A^{-1}A}{1 + u^T A^{-1}u}$$

$$= I - \frac{A^{-1}uu^T}{1 + u^T A^{-1}u} \tag{185}$$

Then,

$$(A + uu^T)^{-1}uu^T = \frac{A^{-1}uu^T}{1 + u^T A^{-1}u}$$

$$(A + uu^T)^{-1} = \frac{A^{-1}}{1 + u^T A^{-1}u} \tag{186}$$

Taking determinants and remembering $|A^{-1}| = \frac{1}{|A|}$ and $|AB| = |BA|$,

$$\frac{1}{|A + uu^T|} = \frac{1}{|A|}\frac{1}{1 + u^T A^{-1}u}$$

$$|A + uu^T| = |A|(1 + u^T A^{-1}u) \tag{187}$$

Appendix C. *Backpropagation for Direct Linear Feedthrough (DLF) Networks*

This appendix will describe how a multilayer perceptron with a DLF structure can be trained using a slightly altered form of backpropagation. Figure 17 in Chapter III illustrates the weight connections in a DLF network.

Let d_j^s denote the desired output value for the jth output node for the sth exemplar and z_j^s denote the actual value for the jth output node for the sth exemplar. For a standard multilayer perceptron, z_j^s is determined by propagating exemplar s using sigmoidal activations at every layer. In the case of a DLF network, the sigmoidal activation at the output layer is not used *and* linear activations are included so that the output of a DLF network is:

$$z_j^s(\text{total}) = z_j^s(\text{net}) + z_j^s(\text{linear}) \tag{188}$$

Let $z_j^s(\text{total}) = \tilde{z}_j^s$. Let total error be defined as

$$E = \sum_s \sum_j \frac{1}{2}(d_j^s - \tilde{z}_j^s)^2 \tag{189}$$

It is this error that is to be minimized by changing the values of the weights. The changing of weights is implemented in traditional backpropagation networks using first-order gradient descent. The updated weight w_{ij}^+ can be written in terms of the previous weight as

$$w_{ij}^+ = w_{ij}^- + \Delta w_{ij} \tag{190}$$

where

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \tag{191}$$

158

To define the weight updates, $\frac{\partial E}{\partial w_{ij}}$ must be investigated further. Let w^2_{IJ} be a particular weight in the upper layer. Then

$$\frac{\partial E}{\partial w^2_{IJ}} = \frac{\partial}{\partial w^2_{IJ}} \left(\sum_s \sum_j \frac{1}{2}(d^s_j - \tilde{z}^s_j)^2 \right)$$

$$= \sum_s \frac{\partial}{\partial w^2_{IJ}} \left(\frac{1}{2}(d^s_J - \tilde{z}^s_J)^2 \right) \tag{192}$$

Note that when the partial is taken inside the sum over j that there is only one node of interest; the sum is therefore eliminated and the subscript is reduced to the particular J that the derivative is being taken with respect to. The derivative then becomes

$$\frac{\partial E}{\partial w^2_{IJ}} = -\sum_s (d^s_J - \tilde{z}^s_J) \frac{\partial \tilde{z}^s_J}{\partial w_{IJ^2}} \tag{193}$$

Now, look at a partial derivative for a particular weight in the second layer w^2_{IJ}:

$$\frac{\partial \tilde{z}^s_J}{\partial w^2_{IJ}} = \frac{\partial \left[\left(\sum_i w^2_{iJ} x^{1s}_i + \xi^2_J \right) + \left(\sum_k w_{kJ} x^s_k + \xi^L \right) \right]}{\partial w^2_{IJ}} \tag{194}$$

where x^{1s}_i is the output of the ith node of the middle layer for the sth exemplar. The derivative becomes

$$\frac{\partial \tilde{z}^s_J}{\partial w^2_{IJ}} = x^{1s}_I \tag{195}$$

and

$$\frac{\partial E}{\partial w^2_{IJ}} = -\sum_s (d^s_J - \tilde{z}^s_J) x^{1s}_I \tag{196}$$

Using the approximation to the total derivative of E found by computing this derivative for a given exemplar and calling this E_s, the calculation required for weight update is

$$\frac{\partial E_s}{\partial w^2_{IJ}} = -(d^s_J - \tilde{z}^s_J) x^{1s}_I \tag{197}$$

Next, look at a particular weight in the lower layer w_{KI}^1 and follow the same approach as above.

$$
\begin{aligned}
\frac{\partial E}{\partial w_{KI}^1} &= \frac{\partial}{\partial w_{KI}^1}\left(\sum_s \sum_j \frac{1}{2}(d_j^s - \tilde{z}_j^s)^2\right) \\
&= \sum_s \sum_j (d_j^s - \tilde{z}_j^s)\frac{\partial}{\partial w_{KI}^1}(d_j^s - \tilde{z}_j^s) \\
&= -\sum_s \sum_j (d_j^s - \tilde{z}_j^s)\frac{\partial}{\partial w_{KI}^1}\left(\sum_i w_{ij}^2 x_i^{1s} + \xi_j^2\right) \quad (198)
\end{aligned}
$$

The only term in the sum over i that is affected by the derivative is when the index $i = I$.

$$
\begin{aligned}
\frac{\partial}{\partial w_{KI}^1}\left(\sum_i w_{ij}^2 x_i^{1s} + \xi_j^2\right) &= \sum_i \frac{\partial}{\partial w_{KI}^1}(x_i^{1s} w_{ij}^2) \\
&= \frac{\partial}{\partial w_{KI}^1}(x_I^{1s} w_{Ij}^2) \\
&= w_{Ij}^2 x_I^{1s}(1 - x_I^{1s})\frac{\partial}{\partial w_{KI}^1}\left(\sum_k x_k^s w_{kI}^1 + \xi_I^1\right) \\
&= w_{Ij}^2 x_I^{1s}(1 - x_I^{1s})x_K^s \quad (199)
\end{aligned}
$$

So the partial derivative of the error with respect to a weight in the first layer becomes

$$
\frac{\partial E}{\partial w_{KI}^1} = -\sum_s \sum_j (d_j^s - \tilde{z}_j^s)w_{Ij}^2 x_I^{1s}(1 - x_I^{1s})x_K^s \quad (200)
$$

Finally, look at a weight associated with one of the linear connections w_{KJ}^L.

$$
\begin{aligned}
\frac{\partial E}{\partial w_{KJ}^L} &= \frac{\partial}{\partial w_{KJ}^L}\left(\sum_s \sum_j \frac{1}{2}(d_j^s - \tilde{z}_j^s)^2\right) \\
&= \sum_s \frac{\partial}{\partial w_{KJ}^L}\left(\frac{1}{2}(d_j^s - \tilde{z}_j^s)^2\right) \\
&= -\sum_s (d_J^s - \tilde{z}_J^s)\frac{\partial \tilde{z}_J^s}{\partial w_{KJ}^L} \quad (201)
\end{aligned}
$$

Now, the last partial derivative becomes

$$
\begin{aligned}
\frac{\partial \tilde{z}_J^s}{\partial w_{KJ}^L} &= \frac{\partial \left[\left(\sum_i w_{iJ}^2 x_i^{1s} + \xi_J^2 \right) + \left(\sum_k w_{kJ}^L x_k^s + \xi^L \right) \right]}{\partial w_{KJ}^L} \\
&= x_k^s
\end{aligned}
\tag{202}
$$

since there is only a single term involving any particular linear weight. So the partial derivative of the error with respect to a linear weight becomes

$$
\frac{\partial E}{\partial w_{KJ}^L} = -\sum_s (d_J^s - \tilde{z}_J^s) x_K^s
\tag{203}
$$

The weight updates outlined above are summarized in the following DLF version of backpropagation:

DLF Backpropagation

1. Initialize weights and biases to small random values.

2. Present training input and desired outputs.

3. Calculate outputs.

4. Adapt weights and biases according to

$$
w_{ij}^+ = w_{ij}^- + \eta \frac{\partial E}{\partial w_{ij}} + \alpha (w_{ij}^- - w_{ij}^{--})
\tag{204}
$$

where

$$
\frac{\partial E}{\partial w_{ij}} = \begin{cases}
(d_j - \tilde{z}_j) x_i^1 & \text{for output node } j \\
x_j^1 (1 - x_j^1) x_i \sum_k (d_k - \tilde{z}_k) w_{jk}^2 & \text{for middle node } j \\
(d_j - \tilde{z}_j) x_i & \text{for linear weight from input } i
\end{cases}
\tag{205}
$$

161

Note that a momentum term has been added with α as the momentum rate, where w_{ij}^- is the old weight value and w_{ij}^{--} is the value of the weight before the last update.

Appendix D. *List of Symbols*

n Number of input nodes, dimensionality of exemplars

N Number of exemplars in the set of interest

m Number of middle nodes

r Number of otuput nodes and/or response functions

$\mathbf{x}_s$ Input vector s, $(s = 1, \ldots, N)$

w_{ki}^1 Weight in first layer connecting input node k to middle node i

      $(k = 1, \ldots, n; i = 1, \ldots m)$

w_{ij}^2 Weight in second layer connecting middle node i to output node j

      $(i = 1, \ldots, m; j = 1, \ldots r)$

ξ_i^1 Bias from first layer to middle node i $(i = 1, \ldots, m)$

ξ_j^2 Bias from second layer to output node j $(j = 1, \ldots, r)$

d_j^s Desired output for output node j with the sth exemplar

z_j^s The multilayer perceptron outputs for the jth output node's

      value with the sth input pattern, or equivalently, the nonlinear regression

      for data vector $\mathbf{x}_s$ with parameters $\mathbf{w}$.

      $\mathbf{z}(\mathbf{x}_s; \mathbf{w})$

$\mathbf{w}$ Vector of all the weights in a neural network $(p \times 1)$

$\hat{\mathbf{w}}$ Estimate of $\mathbf{w}$

$\mathcal{E}_O$ Output Error

$\mathcal{E}_C$ Classification Error

Λ_k Ruck Saliency measure for feature k

$\tilde{\Lambda}_k$ Tarr saliency measure for feature k

$\hat{\Lambda}_k$ Simplified saliency measure for feature k

w_{kj}^L Weight in linear layer of DLF network connecting input node k to

      output node j $(k = 1, \ldots, n; j = 1, \ldots, r)$

ξ^L Bias from input to output layer for DLF networks

F^{DLF} The form that the matrix of first partials takes when a DLF

     network is used

p_n The number of nonlinear parameters in a DLF network

p_l The number of linear parameters in a DLF network

p Number of paramters (weights)

$F.$ Matrix of first partials

$F.(\mathbf{w})$ $N \times p$ matrix of first partials

     $\left\{ \frac{\partial z_s}{\partial w_t} \right\}, s = 1, \ldots, N, t = 1, \ldots, p$

$\boldsymbol{\theta}$ Vector of parameters

$\boldsymbol{\theta}^*$ Represents true parameter vector

$\hat{\boldsymbol{\theta}}$ Represents estimated parameter vector

$\boldsymbol{\theta}$ Shows functional dependedence in equations involving parameter vector

C^{-1} $(F.^T F.)^{-1} = \{ c^{ij} \}$

     The terms c^{ii} are proportional to the variances of the estimated

     parameters $\hat{\theta}_i$ and $c^{ij} (i \neq j)$ are proportional to the covariances

D Design point criterion

$\hat{F}.$ $F.(\hat{\mathbf{w}})$ or $F.(\hat{\boldsymbol{\theta}})$

$(\mathbf{x}_i, y_i)$ Data vectors in the description of nonlinear, single response

     regression model $(i = 1, \ldots, n)$

y_i $f(\mathbf{x}_i; \boldsymbol{\theta}^*) + \varepsilon_i$

     Nonlinear, single response model $(i = 1, \ldots, n)$

$\boldsymbol{\theta}^*$ $p \times 1$ vector of true parameters

$S(\boldsymbol{\theta})$ Error sum of squares function for the single response model

$f_i(\boldsymbol{\theta})$ $f(\mathbf{x}_i; \boldsymbol{\theta})$

     For single response, nonlinear function

$\mathbf{f}(\boldsymbol{\theta})$ $(f_1(\boldsymbol{\theta}), f_2(\boldsymbol{\theta}), \ldots, f_N(\boldsymbol{\theta}))^T$

     Vector of actual responses from the model given $\boldsymbol{\theta}$ and N data vectors

$F.(\boldsymbol{\theta})$ $\frac{\partial f(\boldsymbol{\theta})}{\partial \theta_j} = \left\{ \frac{\partial f_i(\boldsymbol{\theta})}{\partial \theta_j} \right\}$

For the single response case

ε In the single response model, the vector of errors, $E[\varepsilon_i] = 0$ and ε_i are i.i.d with variance σ^2

$\hat{C}$ $\hat{F}.^T \hat{F}.$

$\hat{\sigma}^2$ $\frac{S(\hat{\boldsymbol{\theta}})}{N-p}$

Estimator for σ^2, the error variance

$\mathbf{y}_i$ $\mathbf{f}(\mathbf{x}_i; \boldsymbol{\theta}) + \boldsymbol{\varepsilon}$

Nonlinear multi-response model ($i = 1, \ldots N$)

$\mathbf{y}_i$ $r \times 1$ vector of responses for the ith data vector in the multi-response model

$\boldsymbol{\varepsilon}$ In the multi-response model, i.i.d. with mean $\mathbf{0}$ and variance-covariance matrix Σ

$\mathbf{f}_i(\boldsymbol{\theta})$ $\mathbf{f}(\mathbf{x}_i; \boldsymbol{\theta})$

In multi-response model

$T(\boldsymbol{\theta})$ Error sum of squares function for multi-response model

Ω^{-1} $\Sigma^{-1} \otimes I_N$

Where $\otimes$ is the Kroneker product ($Nr \times Nr$)

$\hat{\sigma}_{rs}$ Estimated value of (r, s) element of Σ

$\mathbf{e}_j$ $\mathbf{y}^{(j)} - \mathbf{f}^{(j)}(\hat{\boldsymbol{\theta}}), j = 1, \ldots r$

Estimated error for response model j

$\hat{\Sigma}$ Estimate of Σ, the variance-covariance matrix of the error

$\hat{W}^{-1}$ Estimate of the variance-covariance matrix of the parameters in the multi-response case ($p \times p$)

$\hat{W}$ $\frac{1}{N} F.^T(\hat{\boldsymbol{\theta}})(\hat{\Sigma}^{-1} \otimes I_N) F.(\hat{\boldsymbol{\theta}})$

$F.(\hat{\boldsymbol{\theta}})$ $\frac{\partial \mathbf{f}(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}^T} = \left\{ \left(\frac{\partial f_r(\hat{\boldsymbol{\theta}})}{\partial \theta_s} \right) \right\}$

Multi-response matrix of first partials

$F_{.j}(\hat{\boldsymbol{\theta}})$ The usual matrix of first partials for the jth model

(See $F_{.}(\hat{\boldsymbol{\theta}})$ for single response model)

η Represents the response for some known function ($\eta = f(\mathbf{x}; \boldsymbol{\theta})$, for example)

$\mathcal{R}$ Region of operability in the feature space

N_0 In a sequential design approach, N_0 is the number of initial observations

$\mathbf{f}_{.i}$ One row of $F_{.}$ for the ith data vector

v_{ij} $\sum_{s=1}^{N} \left(y_{is} - f_i(\mathbf{x}_s; \boldsymbol{\theta})\right) \left(y_{js} - f_j(\mathbf{x}_s; \boldsymbol{\theta})\right), i, j = 1, \ldots, r$

$\mathbf{y}_s^T$ $(y_{1s}, y_{2s}, \ldots y_{ds}), s = 1, \ldots, N$

δ_{is} $y_{is} - f_i(\mathbf{x}_u; \hat{\boldsymbol{\theta}})$

$f_{.is}^{(t)}$ $\left(\frac{\partial f_i(\mathbf{x}_s; \boldsymbol{\theta})}{\partial \theta_t}\right)_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}}$

$(\boldsymbol{\theta}_1 | \boldsymbol{\theta}_2)^T$ Division of parameter vector into subsets where $\boldsymbol{\theta}_1$ is $q \times 1$ and $\boldsymbol{\theta}_2$ is $u \times 1$

$\mathcal{M}_1$ Dot product measure for ranking exemplars

$\mathcal{M}_2$ Saliency measure for ranking exemplars

Bibliography

1. Atkinson, Anthony C. and William G. Hunter. "The Design of Experiments for Parameter Estimation," *Technometrics, 10*: 271-289 (May 1968).

2. Atlas, Les *et al.* "Training Connectionist Networks with Queries and Selective Sampling," in *Advances in Neural Information Processing Systems*, Volume 2, edited by David S. Touretzky. San Mateo CA: Morgan Kaufmann Publishers, 1990.

3. Baum, Eric B. "Neural Net Algorithms That Learn in Polynomial Time from Examples and Queries," *IEEE Transactions on Neural Networks, 2*: 5-19 (January 1991).

4. Beauchamp, K.G. *Applications of Walsh and Related Functions*. New York: Academic Press, 1984.

5. Belue, Capt Lisa M. *An Investigation of Multilayer Perceptrons for Classification*. MS thesis, AFIT/GOR/ENS/92M-02. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1992 (AD-A248086).

6. Belue, Capt Lisa M. and Lt Col Kenneth W. Bauer, Jr. *Methods of Determining Input Features for Multilayer Perceptrons*. Working Paper Series 93, Department of Operational Sciences, School of Engineering, Air Force Institute of Technology, February 1993.

7. Belue, Lisa M. and Kenneth W. Bauer, Jr. "Determining Inputs for Multilayer Perceptrons," *Neurocomputing*, to be published April 1995.

8. Box, George E.P. *et al. Statistics for Experimenters*. New York: John Wiley & Sons, 1978.

9. Box, George E.P. and Norman R. Draper. *Empirical Model-Building and Response Surfaces*. New York: John Wiley & Sons, 1987.

10. ———. "The Bayesian Estimation of Common Parameters from Several Responses," *Biometrika, 52*: 355-365 (December 1965).

11. Box, George E.P. and W.J. Hill. "Discrimination Among Mechanistic Models," *Technometrics, 9*: 57-71 (February 1967).

12. Box, George E.P. and William G. Hunter. "Sequential Design of Experiments for Nonlinear Models," *Proceedings of IBM Scientific Computing Symposium in Statistics*. 113-137, 1965.

13. Box, George E.P and H.L. Lucas. "Design of Experiments in Non-Linear Situations," *Biometrika, 46*: 77-90 (June 1959).

14. Box, M.J. "An experimental design criterion for the precise estimation of a subset of the parameters in a nonlinear model," *Biometrika, 58*: 149-153 (April 1971).

15. Box, M.J. and Norman R. Draper "Estimation and design criteria for multiresponse nonlinear models with non-homogeneous variance," *Applied Statistics, 21*: 13-24 (1972).

16. Cover, Thomas M. "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications to Pattern Recognition," *IEEE Transactions on Electronic Computers, EC-14 (3)*:326-334 (June 1964).

17. de la Maza, M. "Dynamically Adjusting the Number of Hidden Units in a Neural Network," *Artificial Neural Networks*, edited by T. Kohonen, K. Mäkisara, O. Simula and J. Kangas. North Holland, 647-651,1991.

18. Devijver, P.A. and J. Kittler. *Pattern Recognition*. Englewood Cliffs NJ: Prentice Hall, 1982.

19. Dillon, William R. and Matthew Goldstein. *Multivariate Analysis Methods and Applications*. New York: John Wiley & Sons, 1984.

20. Draper, Norman R. and William G. Hunter. "Design of experiments for parameter estimation in multiresponse situations," *Biometrika, 53*: 525-533 (December 1966).

21. ———. "The use of prior distributions in the design of experiments for parameter estimation in nonlinear situations: Multiresponse case," *Biometrika, 54*: 662-665 (1967).

22. Dykstra, Otto. "The Augmentation of Experimental Data to Maximize $|X^T X|$," *Technometrics, 13*: 683-688 (August 1971).

23. Falcon Research and Development (FRD). *Aircraft Fuel Tank Environment/Threat Model for Fire and Explosion Vulnerability Assessment, Volume II. Development of Probabilities and Fire Explosion (U)*. Aeronautical Systems Division Technical Report ASD-TR-77-19, Wright Patterson AFB OH, March 1977.

24. Foley, Donald H. "Considerations of Sample and Feature Size," *IEEE Transactions on Information Theory, 18*: 618-626 (September 1972).

25. Gallant, A. Ronald. *Nonlinear Statistical Models*. New York:John Wiley & Sons, 1987.

26. Gorman, R. Paul and Terrence J. Sejnowski. "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets," *Neural Networks, 1*: 75-89 (1988).

27. Haesloop, Dan and Bradley R. Holt. "Neural Networks for Process Identification," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, III 429-434, 1990.

28. Hecht-Nielsen, Robert. *Neurocomputing*, New York: Addison-Wesley Publishing Company, 1989.

29. Hill, Peter D.H. "D-Optimal Designs for Partially Nonlinear Regression Models," *Technometrics, 22*: 275-276 (May 1980).

30. Hill, William J. and William G. Hunter. "Design of Experiments for Subsets of Parameters," *Technometrics, 16*: 425-434 (August 1974).

31. Hirose, Y., K. Yamashita, and S. Hijiya. "Back-Propagation Algorithm Which Varies the Number of Hidden Units," *Neural Networks, 4*: 61-66, 1991.

168

32. Hopfield, J.J. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Sciences USA, Volume 79*. 2554-2558, April 1982.

33. Huang, S. and Y. Huang. "Bounds on the Number of Hidden Neurons in Multilayer Perceptrons," *IEEE Transactions on Neural Networks 2*: 47-55 (January 1991).

34. Hwang, Jenq-Neng *et al.* "Query-Based Learning Applied to Partially Trained Multilayer Perceptrons," *IEEE Transactions on Neural Networks, 2*: 131-136 (January 1991).

35. Johnson, Mark E. and Christopher J. Nachtsheim. "Some Guidelines for Constructing Exact D-Optimal Designs on Convex Design Spaces," *Technometrics, 25*: 271-277 (August 1983).

36. Kiefer, J. and Wolfowitz, J. "The Equivalence of Two Extremum Problems," *Canadian Journal of Mathematics, 12*: 363-366 (1960).

37. Knight, Cpt Earl E. *Predicting Armor Piercing Incendiary Projectile Effects After Impacting Composite Materials*. MS thesis. AFIT/GOR/ENS/92M-18. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1992 (AD-B162885L).

38. Kung, S.Y. and J.N. Hwang. "An Algebraic Analysis for Optimal Hidden Units Size and Learning Rates in Back-Propagation Learning," *International Conference on Neural Networks*. I 363-370, San Diego, CA, 24-27 July 1988.

39. Lee, Samuel E. and Bradley R. Holt. "Regression Analysis of Spectroscopic Process Data Using A Combined Architecture of Linear and Nonlinear Artificial Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*. IV 549-554, 1992.

40. Lippmann, Richard P. "Pattern Classification Using Neural Networks," *IEEE Communications Magazine 27*: 47-65 (November 1989).

41. MacKay, David J. "Evidence Framework Applied to Classification Networks," *Neural Computation, 4*: 720-736 (1992).

42. Mendenhall, William *et al. Mathematical Statistics with Applications* (Fourth Edition). Boston: PWS-Kent Publishing Company, 1990.

43. Minsky, Marvin Lee and Seymour Papert. *Perceptrons* (Expanded Edition). Cambridge: The MIT Press, 1988.

44. Mitchell, Toby J. "An Algorithm for the Construction of D-Optimal Experimental Designs," *Technometrics,16*: 203-210 (2 May 1974).

45. Myers, Raymond H. *Classical and Modern Regression with Applications*. Boston: PWS-Kent Publishing Company, 1990.

46. Nachtsheim, Christopher J. "Tools for Computer-Aided Design of Experiments," *Journal of Quality Technology, 19*: 132-158 (July 1987).

47. Neter, John *et al. Applied Linear Regression Models* (Second Edition). Homewood IL: Irwin, 1989.

48. Noble, Ben and James W. Daniel. *Applied Linear Algebra* (Second Edition). Englewood Cliffs NJ: Prentice-Hall Inc., 1977.

49. Pettit, Patricia A. *Incendiary Functioning Characteristics of Soviet API Projectiles Impacting Graphite/Epoxy Composite Panels*, Final Report, WRDC-TR-90-3030, Wright-Patterson AFB OH, April 1990.

50. Powell, M.J.D. "On Search Directions for Minimization Algorithms," *Math Programming, 4*: 193-201 (1973).

51. Press, William H. *et al. Numerical Recipes.* Cambridge: Cambridge University Press, 1989.

52. Pukelsheim, Friedrich. *Optimal Design of Experiments.* New York: John Wiley & Sons, 1993.

53. Reinhart, Capt Gregory L. *A FORTRAN Based Learning System Using Multilayer Back-Propagation Neural Network Techniques.* MS thesis, AFIT/GOR/ENS/94M-11. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1994.

54. Reklaitis, G.V. *et al. Engineering Optimization Methods and Applications.* New York: Wiley and Sons, 1983.

55. Rogers, Steven K. and Matthew Kabrisky. *An Introduction to Biological and Artificial Neural Networks for Pattern Recognition.* Bellingham WA: SPIE Optical Engineering Press, 1991.

56. Ruck, Dennis W. *et al.* "Feature selection using a multilayer perceptron," *The Journal of Neural Network Computing, 2*: 40-48 (Fall 1990).

57. Ruck, Capt Dennis W. *Characterization of Multilayer Perceptrons and their Application to Multisensor Automatic Target Detection.* Phd dissertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1990 (AD-A229035).

58. St. John, R.C. and N.R. Draper. "D-Optimality for Regression Designs: A Review," *Technometrics, 17*:15-23 (February 1975).

59. Seber, G.A.F. and C.J. Wild. *Nonlinear Regression.* New York: John Wiley & Sons, 1989.

60. Shields, Paul C. *Elementary Linear Algebra.* New York: Worth Publishers, Inc., 1980.

61. Steppe, Capt Jean M. *Feature and Model Selection in Feedforward Neural Networks.* PhD dissertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1994.

62. Takeuchi, Kei *et al. The Foundations of Multivariate Analysis.* Nwe York: John Wiley & Sons, 1982.

63. Tarr, Capt Greg L. *Multi-Layered Feedforward Neural Networks for Image Segmentation.* Phd dissertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1991 (AD-A243873).

64. Touretzky, David S. and Dean A. Pomerleau. "What's Hidden in the Hidden Layers?" *Byte, 14*: 227-245 (August 1989).

65. Vanderplaats, Garret N. *Numerical Optimization Techniques for Engineering Design with Applications.* New York: McGraw-Hill Book Company, 1984.

66. Walters, Deborah. "Response Mapping Functions: Classification and Analysis of Connectionist Representations," *Proceedings of the IEEE First International Conference on Neural Networks*, III 79-86, June 1987.

67. Weiss, Sholom M. and Casimir A. Kulikowski. *Computer Systems that Learn.* San Mateo CA: Morgan Kaufmann Publishers, 1991.

68. White, Halbert. *Artificial Neural Networks Approximation & Learning Theory.* Cambridge: Blackwell Publishers, 1993.

69. Wijesinha, M.C. and A.I. Khuri. "The Sequential generation of multiresponse D-optimal designs when the variance-covariance matrix is not known," *Comm Statistics and Simulation 16*: 239-258 (1987).

Vita

Captain Lisa M. Belue was born on 21 May 1963 in Port Huron, Michigan. In 1981, she graduated from Port Huron High School and attended Michigan Technological University in Houghton, Michigan. In 1985, she graduated from Michigan Tech with a Bachelor of Science Degree in Mathematics. Her first assignment was as a personnel analyst for The Military Personnel Center at Randolph AFB, Texas. A subsequent assignment in San Antonio was as a future requirements analyst for the Directorate of Development Plans, Headquarters Human Systems Division, Brooks AFB. Captain Belue's next assignment was the Chief, Data Management for the Pacific Air Force's Weapon System Evaluation Program, Clark AB Republic of the Philippines supporting live fire air-to-air missile tests. Captain Belue received her Master's of Science in Operations Research from the Air Force Institute of Technology (AFIT) in 1992. Upon completion of the masters program, she entered into the doctoral program at AFIT.

Permanent address: 1455 Wadhams Road
                       Smiths Creek, Michigan 48074

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE March 1995	3. REPORT TYPE AND DATES COVERED Doctoral Dissertation

4. TITLE AND SUBTITLE
SELECTING OPTIMAL EXPERIMENTS FOR FEEDFORWARD MULTILAYER PERCEPTRONS

5. FUNDING NUMBERS

6. AUTHOR(S)
Lisa M. Belue, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/DS/ENS/95-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
WL/FIVS (Pat Pettit)
1901 10th St.
Wright-Patterson AFB OH 45433-7605

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Where should a researcher conduct experiments to provide training data for a multilayer perceptron? This question is investigated and a statistically-based method for optimally selecting experimental design points for multilayer perceptrons is introduced. Specifically, a criterion is developed based on the size of an estimated confidence ellipsoid for the weights in the multilayer perceptron. This criterion is minimized over a set of exemplars to find optimal design points. Initially, single output networks are examined. An example is used to demonstrate the superiority of optimally selected design points over randomly chosen points and points chosen in a grid pattern. Also, two measures are successfully used to rank the design points in terms of their importance. Two methods are presented to significantly reduce complexity—a distributed linear feedthrough network structure and a weight subset method. Next, multiple output networks are examined. The criterion in this framework becomes more complex and a simplifying technique is employed to judiciously choose desired outputs of the network to produce uncorrelated actual outputs. Finally, the methods described above are integrated and tested on two applications dealing with aircraft survivability. In both cases, simulating the indicated experiments produced a superior multilayer perceptron.

14. SUBJECT TERMS
Neural networks, Pattern recognition, Optimal experiments, Incendiary projectiles, Nonlinear regression

15. NUMBER OF PAGES
172

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL